

Byzantine Multi-Agent Optimization–Part II^{*}

Lili Su

Nitin Vaidya

Department of Electrical and Computer Engineering, and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Email:{lilisu3, nhv}@illinois.edu

Technical Report

July 2015

Abstract

In Part I of this report, we introduced a Byzantine fault-tolerant distributed optimization problem whose goal is to optimize a sum of convex (cost) functions with real-valued scalar input/output. In particular, the goal is to optimize a global cost function $\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} h_i(x)$, where \mathcal{N} is the set of non-faulty agents, and $h_i(x)$ is agent i 's local cost function, which is initially known only to agent i . In general, when some of the agents may be Byzantine faulty, the above goal is unachievable. Therefore, in Part I, we studied a weaker version of the problem whose goal is to generate an output that is an optimum of a function formed as a *convex combination* of local cost functions of the non-faulty agents. We showed that the maximum achievable number of weights (α_i 's) that are bounded away from 0 is $|\mathcal{N}| - f$, where f is the upper bound on the number of Byzantine agents.

In this second part, we introduce a condition-based variant of the original problem over arbitrary directed graphs. Specifically, for a given collection of k input functions $h_1(x), \dots, h_k(x)$, we consider the scenario when the local cost function stored at agent j , denoted by $g_j(x)$, is formed as a *convex combination* of the k input functions $h_1(x), \dots, h_k(x)$. The goal of this condition-based problem is to generate an output that is an optimum of $\frac{1}{k} \sum_{i=1}^k h_i(x)$. Depending on the availability of side information at each agent, two slightly different variants are considered. We show that for a given graph, the problem can indeed be solved despite the presence of faulty agents. In particular, even in the absence of side information at each agent, when adequate *redundancy* is available in the optima of input functions, a distributed algorithm is proposed in which each agent carries minimal state across iterations.

Keywords: Distributed optimization; Byzantine faults; incomplete networks; fault-tolerant computing

1 System Model and Problem Formulation

The system under consideration is synchronous, and consists of n agents connected by an arbitrary directed communication network $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ is the set of n agents, and \mathcal{E} is the set of directed edges between the agents in \mathcal{V} . Up to f of the n agents may be Byzantine faulty.

^{*} This research is supported in part by National Science Foundation awards NSF 1329681 and 1421918. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies or the U.S. government.

Let \mathcal{F} denote the set of faulty agents in a given execution. Agent i can reliably transmit messages to agent j if and only if the directed edge (i, j) is in \mathcal{E} . Each agent can send messages to itself as well, however, for convenience, we *exclude self-loops* from set \mathcal{E} . That is, $(i, i) \notin \mathcal{E}$ for $i \in \mathcal{V}$. With a slight abuse of terminology, we will use the terms *edge* and *link* interchangeably, and use the terms *nodes* and *agents* interchangeably in our presentation.

For each agent i , let N_i^- be the set of agents from which i has incoming edges. That is, $N_i^- = \{j \mid (j, i) \in \mathcal{E}\}$. Similarly, define N_i^+ as the set of agents to which agent i has outgoing edges. That is, $N_i^+ = \{j \mid (i, j) \in \mathcal{E}\}$. Since we exclude self-loops from \mathcal{E} , $i \notin N_i^-$ and $i \notin N_i^+$. However, we note again that each agent can indeed send messages to itself. Agent j is said to be an *incoming neighbor* of agent i , if $j \in N_i^-$. Similarly, j is said to be an *outgoing neighbor* of agent i , if $j \in N_i^+$.

We say that a function $h : \mathbb{R} \rightarrow \mathbb{R}$ is *admissible* if (i) $h(\cdot)$ is convex and L -Lipschitz continuous, and (ii) the set $\text{argmin } h(x)$ containing the optima of $h(\cdot)$ is non-empty and compact (i.e., bounded and closed). Given k admissible input functions $h_1(x), \dots, h_k(x)$, each agent $i \in \mathcal{V}$ is initially provided with a local cost function $g_i(\cdot)$ of the form

$$g_i(x) = \mathbf{A}_{1i}h_1(x) + \mathbf{A}_{2i}h_2(x) + \dots + \mathbf{A}_{ki}h_k(x),$$

where $\mathbf{A}_{ji} \geq 0$ and $\sum_{j=1}^k \mathbf{A}_{ji} = 1$ for all $i \in \mathcal{V}$ and all $j = 1, \dots, k$. Compactly, we have $\mathbf{g}(x) = \mathbf{h}(x)\mathbf{A}$, where $\mathbf{h}(x) = [h_1(x), h_2(x), \dots, h_k(x)]$, $\mathbf{g}(x) = [g_1(x), g_2(x), \dots, g_n(x)]$ and $\mathbf{A} \in \mathbb{R}^{k \times n}$. Our problem formulation is motivated by the work on condition-based consensus [15, 6, 10], where the inputs of the agents are restricted to be within some acceptable set.

Each agent i maintains state x_i , with $x_i(t)$ denoting the local estimate of the optimal x , computed by node i at the *end* of the t -th iteration of the algorithm, with $x_i(0)$ denoting its initial local estimate. At the *start* of the t -th iteration ($t > 0$), the local estimate of agent i is $x_i(t-1)$. The algorithms of interest will require each agent i to perform the following three steps in iteration t , where $t > 0$. Note that the faulty agents may deviate from this specification. Since each $h_j(\cdot)$ is convex and L -Lipschitz continuous, and $\sum_{j=1}^k \mathbf{A}_{ji} = 1$, it follows that each $g_i(\cdot)$ is also convex and L -Lipschitz continuous. Note that the formulation allows $n < k$ as well as $n \geq k$. The matrix \mathbf{A} is termed as a *job assignment matrix*. The goal here is to develop algorithms that output $x_i = \tilde{x}$ at each non-faulty agent i such that

$$\tilde{x} \in \text{argmin } h(x) = \frac{1}{k} \sum_{j=1}^k h_j(x). \quad (1)$$

That is, we are interested in developing algorithms in which the local estimate of each non-faulty agent will eventually reach consensus, and the consensus value is an optimum of function $h(\cdot)$.

Let $X_j = \text{argmin } h_j(x)$ for all $j = 1, \dots, k$, and let $X = \text{argmin } h(x)$. For ease of future reference, we refer to the above optimization problem 1 as Problem (1). Problem 1 is said to be solvable if there exists an algorithm that outputs $\tilde{x} \in \text{argmin } h(x)$ at each non-faulty agent i for any collection of k admissible functions. Problem 1 can be further formulated differently depending on whether each non-faulty agent i knows the assignment matrix \mathbf{A} or not. We refer to the formulation where the agents know matrix \mathbf{A} as condition-based Byzantine multi-agent optimization **with** side information; otherwise the problem is called condition-based Byzantine multi-agent optimization **without** side information.

Our formulation is more general than the common formulation adopted in [8, 17, 18, 21, 24, 25], in which $f = 0$ and the assignment matrix $\mathbf{A} = \mathbf{I}_k$ (identity matrix) is considered. Despite the elegance

of the algorithms proposed in [8,17,18,21,24,25], none of these algorithms work in the presence of Byzantine agents when $f \geq 1$ and $\mathbf{A} = \mathbf{I}_k$. Informally speaking, this is because under $\mathbf{g}(x) = \mathbf{h}(x)\mathbf{I}_k = \mathbf{h}(x)$ assignment, the information about the input function $h_i(x)$ is exclusively known to agent i in the system. If agent i is faulty and misbehaves, or crashes at the beginning of an execution, then the information about $h_i(x)$ is not accessible to the non-faulty agents. When $\sum_{j=1}^k h_j(x)$ and $\sum_{j=1, j \neq i}^k h_j(x)$ do not have common optima, there does not exist a correct algorithm. A stronger impossibility result is presented next, which is proved in Part I of our work [23].

Theorem 1. [23] *Problem 1 is not solvable when $f \geq 1$ and $\mathbf{A} = \mathbf{I}_k$.*

In contrast, function redundancy can be added to the system by applying a properly chosen job assignment matrix \mathbf{A} to $\mathbf{h}(x)$. For example, suppose $k = 2$, $f = 1$ and the optimal sets of functions $h_1(x)$ and $h_2(x)$ are $[-1, 0]$ and $[0, 1]$, respectively. Let $\mathbf{g}(x) = \mathbf{h}(x)\mathbf{G}$, where \mathbf{G} is a generator matrix of a repetition code with $d = 2f + 1 = 3$. Informally speaking, by applying linear code \mathbf{G} on input functions $\mathbf{h}(x)$, i.e., $\mathbf{g}(x) = \mathbf{h}(x)\mathbf{G}$, the Byzantine agents' ability in hiding information about input functions can be weakened. This observation and Theorem 1 together justify our problem formulation.

Contributions: We introduce a condition-based approach to Byzantine multi-agent optimization problem. Two slightly different variants are considered: condition-based Byzantine multi-agent optimization with side information and condition-based Byzantine multi-agent optimization without side information. For the former, when side information is available at each agent, a decoding-based algorithm is proposed, assuming that each input function is differentiable. This algorithm combines the gradient method with the decoding procedure introduced in [4] (namely matrix \mathbf{A}). With such a decoding subroutine, our algorithm essentially performs the gradient method, where gradient computation is performed distributedly over the multi-agent system. When side information is not available at each agent, we propose a simple consensus-based algorithm in which each agent carries minimal state across iterations. This consensus-based algorithm solves Problem 1 under the additional assumption over input functions that all input functions share at least one common optimum.

Organization: The rest of the report is organized as follows. Related work is summarized in Section 2. Condition-based Byzantine multi-agent optimization with side information is analyzed in Section 3, where each agent knows the assignment matrix \mathbf{A} . Section 4 is devoted to the case when each agent does not know \mathbf{A} . Section 5 concludes the report.

2 Related Work

Fault-tolerant consensus [19] is closely related to the optimization problem considered in this report. There is a significant body of work on fault-tolerant consensus, including [7,6,14,9,12,27,10]. Two variants that are most relevant to the algorithms in this report are *iterative approximate Byzantine consensus* [9,12,27] and *condition-based consensus* [15,6,10]. Iterative approximate consensus requires that the agents agree with each other only approximately, using local communication and maintaining *minimal* state across iterations. Condition-based consensus [15] restricts the inputs of the agents to be within some acceptable set. [6] showed that if a condition (the set of allowable system inputs) is *f-acceptable*, then consensus can be achieved in the presence of up to f crash failures over complete graphs. A connection between asynchronous consensus and error-correcting codes (ECC) was established in [10], observing that crash failures and Byzantine failures correspond

to erasures and substitution errors, respectively, in ECCs. Condition-based approach can also be used in synchronous system to speed up the agreement [16,15].

Convex optimization, including distributed convex optimization, also has a long history [2]. Primal and dual decomposition methods that lend themselves naturally to a distributed paradigm are well-known [3]. There has been significant research on a variant of distributed optimization problem [8,17,18,21,24,25], in which the global objective $h(x)$ is a summation of n convex functions, i.e., $h(x) = \sum_{j=1}^n h_j(x)$, with function $h_j(x)$ being known to the j -th agent. The need for robustness for distributed optimization problems has received some attentions recently [8,17,21]. In particular, Ram et al. [21] studied the scenario when each component function is known partially (with stochastic errors) to an agent, Duchi et al. [8] and Nedic et al. [17] investigated the impact of random communication link failures and time-varying communication topology. Duchi et al. [8] assumed that each realizable link failure pattern considered in [8] is assumed to admit a doubly-stochastic matrix which governs the evolution dynamics of local estimates of the optimum. The doubly-stochastic requirement is relaxed in [17], using the push-sum technique used in [24]. In contrast, we consider the system in which up to f agents may be Byzantine, i.e., up to f agents may be adversarial and try to mislead the system to function improperly. We are not aware of the existence of results obtained in this report.

In other related work, significant attempts have been made to solve the problem of distributed hypothesis testing in the presence of Byzantine attacks [11,29,13], where Byzantine sensors may transmit fictitious observations aimed at confusing the decision maker to arrive at a judgment that is in contrast with the true underlying distribution. Consensus based variant of distributed event detection, where a centralized data fusion center does not exist, is considered in [11]. In contrast, in this paper, we focus on the Byzantine attacks on the multi-agent optimization problem.

3 Condition-based Byzantine multi-agent optimization with side information

In this section we consider condition-based Byzantine multi-agent optimization with side information, where each agent knows the assignment matrix \mathbf{A} . Let $\{\alpha(t)\}_{t=0}^{\infty}$ be a sequence of step sizes. A simple decoding-based algorithm, Algorithm 1, formally presented below, works in an iterative fashion. Recall that $x_i(0)$ is the initial state of local estimate for each non-faulty agent $i \in \mathcal{V} - \mathcal{F}$, and $G(\mathcal{V}, \mathcal{E})$ is the underlying communication graph. Without loss of generality, we assume that $x_i(0) = x_0$ for $i \in \mathcal{V} - \mathcal{F}$ and some arbitrary but fixed $x_0 \in \mathbb{R}$. Otherwise, we can add an additional initialization step to guarantee identical “initial state” using an arbitrary exact consensus algorithm. Let $x_i(t)$ be the local estimate of an optimum in X , computed by node i at the *end* of the t -th iteration of the algorithm. At the *start* of the t -th iteration ($t > 0$), the local estimate of agent i is $x_i(t-1)$.

For Algorithm 1 to work, we assume that each input function $h_i(\cdot)$ is differentiable. Consequently, the local objective $g_i(\cdot)$ is also differentiable for each $i \in \mathcal{V}$. Let $\mathbf{A} \in \mathbb{R}^{k \times n}$ be a matrix that can corrects up to f arbitrary entry-wise errors in [4]. At iteration t , each non-faulty agent i computes the gradient of $g_i(t)$ at $x_i(t-1)$. Let $\mathbf{d}(t)$ be the k -dimensional vector of the gradients of the k input functions at $x_i(t-1)$, where $i \in \mathcal{V} - \mathcal{F}$. For the j -th entry in $\mathbf{d}(t)$, i.e., $\mathbf{d}_j(t)$, it holds that $\mathbf{d}_j(t) = h'_j(x_i(t-1))$. Later we will show that $x_i(t-1) = x_j(t-1)$ for all $i, j \in \mathcal{V} - \mathcal{F}$. Thus $\mathbf{d}(t)$ is well-defined. In addition, we assume the structure of the underlying graph $G(\mathcal{V}, \mathcal{E})$ admits Byzantine broadcast. For instance, when $G(\mathcal{V}, \mathcal{E})$ is undirected, for a correct Byzantine broadcast algorithm to exist, node connectivity of $G(\mathcal{V}, \mathcal{E})$ is at least $2f + 1$.

Algorithm 1:

Steps to be performed by agent $i \in \mathcal{V}$ in iteration $t \geq 0$.

Initialization: $x_i(0) \leftarrow x_0$.

1. *Transmit step:* Compute $g'_i(x_i(t-1))$, the gradient of $g_i(\cdot)$ at $x_i(t-1)$, and perform Byzantine broadcast of $g'_i(x_i(t-1))$ to all agents.
2. *Receive step:* Receive gradients from all other agents. Let $\mathbf{y}^i(t)$ be a n -dimensional vector of received gradients, with $\mathbf{y}_j^i(t)$ being the value received from agent j . If $j \in \mathcal{V} - \mathcal{F}$, then $\mathbf{y}_j^i(t) = g'_j(x_j(t-1))$.
3. *Gradient Decoding step:* Perform the decoding procedure in [4] to recover

$$\mathbf{d}(t) = [h'_1(x_i(t-1)), \dots, h'_k(x_i(t-1))]^T.$$

4. *Update step:* Update its local estimate as follows.

$$x_i(t) = x_i(t-1) - \alpha(t-1) \sum_{j=1}^k h'_j(x_i(t-1)). \quad (2)$$

At iteration $t = 1$, each non-faulty agent i computes $g'_i(x_i(0))$ —the gradient of $g_i(\cdot)$ at the current estimate $x_i(0) = x_0$, and performs Byzantine broadcast of $g'_i(x_0)$. Note that a faulty agent p , instead of $g'_p(x_0)$, may perform Byzantine broadcast of some arbitrary value to other agents. Recall that $\mathbf{y}^i(1) \in \mathbb{R}^n$ is a n -dimensional real vector, with $\mathbf{y}_j^i(1)$ be the value received from agent j at iteration 1. Since $\mathbf{g}(\cdot) = \mathbf{h}(\cdot)\mathbf{A}$, then we can write $\mathbf{y}^i(1)$ as $\mathbf{y}^i(1) = \mathbf{d}(1)\mathbf{A} + \mathbf{e}^i(1)$, where $\mathbf{e}^i(1)$ corresponds to the errors induced by the faulty agents. Let p be a nonzero entry in $\mathbf{e}^i(1)$, it should be noted that $\mathbf{e}_p^i(1)$ can be arbitrarily away from 0. Since messages/values are transmitted via Byzantine broadcast, it holds that $\mathbf{e}^i(1) = \mathbf{e}^{i'}(1)$ for all $i, i' \in \mathcal{V} - \mathcal{F}$. Consequently, we have $\mathbf{y}^i(1) = \mathbf{y}^{i'}(1)$ for all $i, i' \in \mathcal{V} - \mathcal{F}$. For each $i \in \mathcal{V} - \mathcal{F}$, $\mathbf{d}(1)$ can be recovered using the decoding procedure in [4]. By the updating function (2), we know $x_i(1) = x_j(1)$ for all $i, j \in \mathcal{V} - \mathcal{F}$. Inductively, it can be shown that $x_i(t) = x_j(t)$ for all $i, j \in \mathcal{V} - \mathcal{F}$, and for all $t \geq 0$. Thus, $\mathbf{d}(t)$ is well-defined for each $t \geq 0$. The remaining correctness proof of Algorithm 1 follows directly from the standard gradient method convergence analysis for convex objective.

Due to the use of Byzantine broadcast, the communication load in Algorithm 1 is high. The communication cost can be reduced by using a matrix \mathbf{A} that has stronger error-correction ability. In general, there is some tradeoff among the communication cost, the graph structure and the error-correcting capability of \mathbf{A} . Our main focus of this paper is the case when no side-information is available at each agent, thus we do not pursue this tradeoff further.

4 Condition-based Byzantine multi-agent optimization without side information

In this section, we consider the scenario when side information about the assignment matrix \mathbf{A} is not known to each agent. We will classify the collection of input functions into three classes depending on the level of redundancy in the input function solutions. For functions with adequate redundancy in their optima, a simple consensus-based algorithm, named Algorithm 2, is proposed. Although Algorithm 2, at least in its current form, only works for a restricted class of input functions, it is

more efficient in terms of both memory and local computation, compared to Algorithm 1. We leave the adaptation of Algorithm 2 to the general input functions as future work.

The job assignment matrices used in this section are characterized by *sparsity parameter*—a new property (introduced in this report) over matrices.

4.1 Classification of input functions collections

Recall that protective function redundancy is added to the system by applying a proper matrix \mathbf{A} to $\mathbf{h}(\cdot)$, i.e., $\mathbf{g}(\cdot) = \mathbf{h}(\cdot)\mathbf{A}$. In Algorithm 1 sufficient redundancy is added to the system such that Algorithm 1 works for any collection of input functions. However, for some collection of input functions, such function redundancy may not be necessary. Consider the case when all k input functions are strictly convex and have the same optimum, i.e., $X_i = \{x^*\}$ for some x^* and for all $i = 1, \dots, k$. In addition, the agents know that $X_i = X_j$ and $|X_i| = 1$ for all $i, j \in \mathcal{V}$. It can be checked that $h(x) = \frac{1}{k} \sum_{j=1}^k h_j(x)$ is also strictly convex and $X = \{x^*\}$. Even if there is no redundant agents in the system and no redundancy added when applying \mathbf{A} , i.e., $\mathbf{A} = \mathbf{I}_k$, Problem 1 can be solved trivially by requiring each non-faulty agent to minimize its own local objective $h_i(x)$ individually without exchanging any information with other agents.

Informally speaking, as suggested by the above example, the optimal sets of the given input functions may themselves have redundancy. For ease of further reference, we term this redundancy as *solution redundancy*. Closer examination reveals that the collections of input functions can be categorized into three classes according to solution redundancy.

Case 1: The k input functions are strictly convex and $X_i = \{x^*\}$ for all $i = 1, \dots, k$, and the agents know that $X_i = X_j$ and $|X_i| = 1$ for all $i, j \in \mathcal{V}$;

Case 2: The k input functions share at least one common optimum, i.e., $\cap_{i=1}^k X_i \neq \emptyset$, and the agents know that $\cap_{i=1}^k X_i \neq \emptyset$;

Case 3: The k input functions share no optima, i.e., $\cap_{i=1}^k X_i = \emptyset$.

If the collection of k input functions belongs to Case 1 or Case 2, we refer to this scenario as *solution-redundant* functions; similarly, we refer to the collection of k functions that falls within Case 3 as *solution-independent* functions. When the given collection of input functions fits Case 2 or Case 3 (but not Case 1), information exchange among agents is in general required in order to achieve asymptotic consensus over local estimates of non-faulty agents.

In this section, we are particularly interested in the family of algorithms of the following structure.

4.2 Algorithm Structure

Recall that each agent i maintains state x_i , with $x_i(t)$ denoting the local estimate of an optimum in X , computed by node i at the *end* of the t -th iteration of the algorithm, with $x_i(0)$ denoting its initial local estimate. At the *start* of the t -th iteration ($t > 0$), the local estimate of agent i is $x_i(t-1)$. The algorithms of interest will require each agent i to perform the following three steps in iteration t , where $t > 0$. Note that the faulty agents may deviate from this specification.

1. *Transmit step*: Transmit message $m_i(t)$ on all outgoing edges (to agents in N_i^+).
2. *Receive step*: Receive messages on all incoming edges (from agents in N_i^-). Denote by $r_i(t)$ the vector of messages received from its neighbors.

3. *Update step*: Agent i updates its local estimate using a transition function Z_i ,

$$x_i(t) = Z_i(r_i(t), x_i(t-1), g_i(\cdot)), \quad (3)$$

where Z_i is a part of the specification of the algorithm.

The evolution of local estimate at agent i is governed by the update function defined in (3). Note that $x_i(t)$ only depends on local objective $g_i(\cdot)$, $x_i(t-1)$ and $r_i(t)$ —the messages collected by agent i in the receive step of iteration t . No other information collected in any of the previous iteration will affect the update step in iteration t . Intuitively speaking, non-faulty agent i is assumed to have no memory across iterations except x_i . Note that the information available at each non-faulty node $i \in \mathcal{V} - \mathcal{F}$ is the local estimate $x_i(t-1)$ and the local objective $g_i(\cdot)$. Thus, the message $m_i(t)$ is a function of $x_i(t-1)$ and $g_i(\cdot)$ only, i.e.,

$$m_i = F_i(x_i(t-1), g_i(\cdot)).$$

An algorithm is said to be correct (1) if $\lim_{t \rightarrow \infty} |x_i(t) - x_j(t)| = 0$ and $\lim_{t \rightarrow \infty} x_j(t) \in X$, for all initial states $x_j(0)$ and for all $i, j \in \mathcal{V} - \mathcal{F}$, and (2) if there exists a finite t_0 such that $x_i(t_0) = x_j(t_0)$ and $x_i(t_0) \in X$ for all $i, j \in \mathcal{V} - \mathcal{F}$, then $x_i(t) = x_j(t)$ for all $i \in \mathcal{V} - \mathcal{F}$ and for all $t \geq t_0$.

Case 1 above is a special form of Case 2. For Case 1, where $h_j(x)$'s are strictly convex and have the same optimum, the problem can be solved trivially. However, for Case 2 in general, the redundancy that is necessary may depend on the underlying graph structure. Henceforth, we consider the scenario when the input functions falls in Case 2. Note that Theorem 1 still holds when restricting to Case 2 input functions. Next, we introduce the notion of sparsity parameter of a job assignment matrix, and characterize the tradeoff between the sparsity parameter and the necessary and sufficient condition, for a correct algorithm to exist.

Definition 1. *Given a job assignment matrix \mathbf{A} , the sparsity parameter of \mathbf{A} , denoted by $sp(\mathbf{A})$, is the smallest integer such that the sum vector of any $sp(\mathbf{A})$ columns of \mathbf{A} is component-wise positive, i.e., every coordinate of the sum vector is positive. In particular, if the sum vector of all columns of \mathbf{A} is not component-wise positive, then $sp(\mathbf{A}) \triangleq n + 1$ by convention.*

Recall that $\mathbf{A} \geq \mathbf{0}$ is a nonnegative matrix, $sp(\mathbf{A}) \triangleq n + 1$ implies that there exists a row in \mathbf{A} that contains only zeros. The following lemma presents a lower bound on the number of nonzero elements in a row of \mathbf{A} , given that $sp(\mathbf{A}) = k'$.

Lemma 1. *Given an assignment matrix \mathbf{A} , its sparsity parameter $sp(\mathbf{A}) = k'$ if and only if there are at most $k' - 1$ zero entries in each row of \mathbf{A} and there exists one row that contains exactly $k' - 1$ zero entries.*

Lemma 1 is proved in Appendix B.

The sparsest assignment matrix \mathbf{A} with $sp(\mathbf{A}) = k'$ can be constructed by choosing arbitrary $k' - 1$ entries in each row to be zero. By the proof of Lemma 1, it can be checked that the sparsity parameter of the obtained matrix \mathbf{A} is k' . In addition, the total number of non-zero entries in \mathbf{A} is $(n - k' + 1)k$.

4.3 Terminology of Consensus

Our condition is based on characterizing a special of subgraphs of $G(\mathcal{V}, \mathcal{E})$, termed by reduced graph [27], formally defined below.

Definition 2. [27] *For a given graph $G(\mathcal{V}, \mathcal{E})$, a reduced graph \mathcal{H} is a subgraph of $G(\mathcal{V}, \mathcal{E})$ obtained by (i) removing all the faulty agents from \mathcal{V} along with their edges; (ii) removing any additional up to f incoming edges at each non-faulty agent.*

Let us denote the collection of all the reduced graphs for a given $G(\mathcal{V}, \mathcal{E})$ by $R_{\mathcal{F}}$. Thus, $\mathcal{V} - \mathcal{F}$ is the set of agents in each element in $R_{\mathcal{F}}$. Let $\tau = |R_{\mathcal{F}}|$. It is easy to see that τ depends on \mathcal{F} as well as the underlying network $G(\mathcal{V}, \mathcal{E})$, and it is finite.

Definition 3. *A source component¹ S of a given graph $G(\mathcal{V}, \mathcal{E})$ is the collection of agents each of which has a directed path to every other agent in $G(\mathcal{V}, \mathcal{E})$.*

It can be easily checked that if the source component S , if any, is a strongly connected component in $G(\mathcal{V}, \mathcal{E})$. In addition, a graph contains at most one source component.

4.4 Necessary Condition

We now present a necessary condition on the underlying communication graph $G(\mathcal{V}, \mathcal{E})$ for solving Problem 1. Our necessary condition is based on characterizing the connectivity of each reduced graph of $G(\mathcal{V}, \mathcal{E})$.

Theorem 2. *Given a graph $G(\mathcal{V}, \mathcal{E})$, if there exists a correct algorithm that can solve Problem 1 when the agents do not have knowledge of the matrix, under any assignment matrix \mathbf{A} for any k solution-redundant input functions, then a source component must exist containing at least $\max\{f+1, sp(\mathbf{A})\}$ nodes.*

The proof of Theorem 2 can be found in Appendix B.

For future reference, we term the necessary condition in Theorem 2 as Condition 1. Condition 1 also implies a lower bound on the number of agents needed, stated below.

Corollary 1. *For a given graph $G(\mathcal{V}, \mathcal{E})$, if Condition 1 is true, then $n \geq \max\{sp(\mathbf{A})+2f, 3f+1\}$.*

It can be shown that this lower bound is indeed tight. For instance, the complete graph of size $sp(\mathbf{A})+2f$, denoted by $K_{sp(\mathbf{A})+2f}$. It can be easily proved by contradiction that $K_{sp(\mathbf{A})+2f}$ satisfies Condition 1. The proof of Corollary 1 is presented in Appendix B.

4.5 Sufficiency of Condition 1

Let $\{\alpha(t)\}_{t=0}^{\infty}$ be a sequence of stepsizes such that $\alpha(t) \leq \alpha(t+1)$ for all $t \geq 0$, $\sum_{t=0}^{\infty} \alpha(t) = \infty$, and $\sum_{t=0}^{\infty} \alpha^2(t) < \infty$. We show that Condition 1 is also sufficient. Let $\phi = |\mathcal{F}|$. Thus $\phi \leq f$. Without loss of generality, let us assume that the non-faulty agents are indexed as 1 to $n - \phi$. Recall that the system is synchronous. If a non-faulty agent does not receive an expected message from an incoming neighbor (in the *Receive step* below), then that message is assumed to have some default value. With the exception of the update step (4) below, the algorithm is similar to the consensus algorithms in [27, 26, 18].

¹ The definition of a source is different from [28].

Algorithm 2

Steps to be performed by agent $i \in \mathcal{V} - \mathcal{F}$ in the t -th iteration:

1. *Transmit step*: Transmit current state $x_i(t-1)$ on all outgoing edges.
2. *Receive step*: Receive values on all incoming edges. These values form multiset² $r_i(t)$ of size $|N_i^-|$.
3. *Update step*: Sort the values in $r_i(t)$ in an increasing order, and eliminate the smallest f values, and the largest f values (breaking ties arbitrarily). Let $N_i^*(t)$ denote the identifiers of agents from whom the remaining $|N_i^-| - 2f$ values were received, and let w_j denote the value received from agent $j \in N_i^*(t)$. For convenience, define $w_i = x_i(t-1)$.³ Update its state as follows.

$$x_i(t) = \sum_{j \in \{i\} \cup N_i^*(t)} a_i w_j - \alpha(t-1) d_i(t-1), \quad (4)$$

where $a_i = \frac{1}{|N_i^*(t)|+1}$ and $d_i(t-1)$ is a gradient of agent i 's objective function $g_i(\cdot)$ at $x = x_i(t-1)$.

Recall that $i \notin N_i^*(t)$ because $(i, i) \notin \mathcal{E}$. The “weight” of each term on the right-hand side of (4) is a_i , and these weights add to 1. Observe that $0 < a_i \leq 1$. Let $\mathbf{x} \in \mathbb{R}^{n \times \phi}$, be a real vector of dimension $n - \phi$, with x_i being the local estimate of agent $i, \forall i \in \mathcal{V} - \mathcal{F}$. Thus, $\mathbf{x}(t)$ is a vector of the local estimates of non-faulty agents at iteration t .

Since $G(\mathcal{V}, \mathcal{E})$ satisfies Condition 1, as shown in [26], the updates of $\mathbf{x} \in \mathbb{R}^{n-\phi}$ in each iteration can be written compactly in a matrix form.

$$\mathbf{x}(t+1) = \mathbf{M}(t)\mathbf{x}(t) - \alpha(t)\mathbf{d}(t). \quad (5)$$

The construction of $\mathbf{M}(t)$ and relevant properties are given in [26] and are also presented in Appendix C for completeness. Let $\mathcal{H} \in R_{\mathcal{F}}$ be a reduced graph of the given graph $G(\mathcal{V}, \mathcal{E})$ with \mathbf{H} as adjacency matrix. It is shown that in every iteration t , and for every $\mathbf{M}(t)$, there exists a reduced graph $\mathcal{H}(t) \in R_{\mathcal{F}}$ with adjacency matrix $\mathbf{H}(t)$ such that

$$\mathbf{M}(t) \geq \beta \mathbf{H}(t), \quad (6)$$

where $0 < \beta < 1$ is a constant. The definition of β can be found in [26]. Equation (5) can be further expanded out as

$$\mathbf{x}(t+1) = \mathbf{\Phi}(t, 0)\mathbf{x}(0) - \sum_{r=1}^{t+1} \alpha(r-1)\mathbf{\Phi}(t, r)\mathbf{d}(r-1), \quad (7)$$

where $\mathbf{\Phi}(t, r) = \mathbf{M}(t)\mathbf{M}(t-1)\dots\mathbf{M}(r)$ and by convention $\mathbf{\Phi}(t, t) = \mathbf{M}(t)$ and $\mathbf{\Phi}(t, t+1) = \mathbf{I}_{n-\phi}$, the identity matrix. Note that $\mathbf{\Phi}(t, r)$ is a backward product (i.e., therein index decrease from left to right in the product).

² In a multiset, multiple instances of an element is allowed. For instance, $\{1, 1, 2\}$ is a multiset.

³ Observe that if $j \in \{i\} \cup N_i^*(t)$ is non-faulty, then $w_j = x_j(t-1)$.

Convergence of the Transition Matrices $\Phi(t, r)$ It can be seen from (7) that the evolution of estimates of non-faulty agents $\mathbf{x}(t)$ is determined by the backward product $\Phi(t, r)$. Thus, we first characterize the evolutionary properties and limiting behaviors of the backward product $\Phi(t, r)$, assuming that the given $G(\mathcal{V}, \mathcal{E})$ satisfies Condition 1.

Let $k' = sp(\mathbf{A})$. The following lemma describes the structural property of $\Phi(t, r)$ for sufficient large t . For a given r , Lemma 2 states that all non-faulty agents will be influenced by at least $\max\{k', f + 1\}$ common non-faulty agents, and this set of influencing agents may depend on r . Proof of Lemma 2 can be found in Appendix D.

Lemma 2. *There are at least $\max\{sp(\mathbf{A}), f + 1\}$ columns in $\Phi(r + \nu - 1, r)$ that are lower bounded by $\beta^\nu \mathbf{1}$ component-wise for all r , where $\mathbf{1} \in \mathbb{R}^{n-\phi}$ is an all one column vector of dimension $n - \phi$.*

Using coefficients of ergodicity theorem, it is showed in [26] that if the given graph $G(\mathcal{V}, \mathcal{E})$ satisfies Condition 1, then $\Phi(t, r)$ is weak-ergodic. Moreover, because weak-ergodicity is equivalent to strong-ergodicity for backward product of stochastic matrices [5], as $t \rightarrow \infty$ the limit of $\Phi(t, r)$ exists

$$\lim_{t \geq r, t \rightarrow \infty} \Phi(t, r) = \mathbf{1}\pi(r), \quad (8)$$

where $\pi(r) \in \mathbb{R}^{n-\phi}$ is a row stochastic vector (may depend on r). It is shown, using ergodic coefficients, in [1] that the rate of the convergence in (8) is exponential, as formally stated in Theorem 3. Recall that $\tau = |R_{\mathcal{F}}|$, $n - \phi$ is the total number of non-faulty agents, and $0 < \beta < 1$ is a constant for which (6) holds.

Theorem 3. [1] *Let $\nu = \tau(n - \phi)$ and $\gamma = 1 - \beta^\nu$. For any sequence $\Phi(t, r)$,*

$$|\Phi_{ij}(t, r) - \pi_j(r)| \leq \gamma^{\lceil \frac{t-r+1}{\nu} \rceil}, \quad (9)$$

for all $t \geq r$.

Our next lemma is an immediate consequence of Lemma 2 and the convergence of $\Phi(t, r)$, stated in (8).

Lemma 3. *For any fixed r , there exists a subset $\mathcal{I}_r \subseteq \mathcal{V} - \mathcal{F}$ such that $|\mathcal{I}_r| \geq \max\{sp(\mathbf{A}), f + 1\}$ and for each $i \in \mathcal{I}_r$,*

$$\pi_i(r) \geq \beta^\nu.$$

The proof of Lemma 3 can be found in Appendix D.

Convergence Analysis of Algorithm 2 Here, we study the convergence behavior of Algorithm 2. The structure of our convergence proof is rather standard, which is also adopted in [8, 18, 21, 24, 25]. We have shown that the evolution dynamics of $\mathbf{x}(t)$ is captured by (5) and (7). Suppose that all agents, both non-faulty agents and faulty agents cease computing $d_i(t)$ after some time \bar{t} , i.e., after \bar{t} subgradient is replaced by 0.

Let $\{\bar{\mathbf{x}}(t)\}$ be the sequences of local estimates generated by the non-faulty agents in this case. From (7) we get

$$\bar{\mathbf{x}}(t) = \mathbf{x}(t),$$

for all $t \leq \bar{t}$. From (5) and (7), we have for all $s \geq 0$, it holds that

$$\bar{\mathbf{x}}(\bar{t} + s + 1) = \mathbf{\Phi}(t, 0)\mathbf{x}(0) - \sum_{r=1}^{\bar{t}} \alpha(r-1) \mathbf{\Phi}(\bar{t} + s, r) \mathbf{d}(r-1). \quad (10)$$

Note that the summation in RHS of (10) is over \bar{t} terms since all agents cease computing $d_j(t)$ starting from iteration \bar{t} . As $s \rightarrow \infty$, we have

$$\begin{aligned} \lim_{s \rightarrow \infty} \bar{\mathbf{x}}(\bar{t} + s + 1) &= \lim_{s \rightarrow \infty} \mathbf{\Phi}(t, 0)\mathbf{x}(0) - \sum_{r=1}^{\bar{t}} \alpha(r-1) \mathbf{\Phi}(\bar{t} + s, r) \mathbf{d}(r-1) \\ &= \lim_{s \rightarrow \infty} \mathbf{\Phi}(t, 0)\mathbf{x}(0) - \left(\sum_{r=1}^{\bar{t}} \alpha(r-1) \lim_{s \rightarrow \infty} \mathbf{\Phi}(\bar{t} + s, r) \mathbf{d}(r-1) \right) \\ &= \mathbf{1}\pi(0)\mathbf{x}(0) - \left(\sum_{r=1}^{\bar{t}} \alpha(r-1) \mathbf{1}\pi(r) \mathbf{d}(r-1) \right) \\ &= \left(\langle \pi(0), \mathbf{x}(0) \rangle - \sum_{r=1}^{\bar{t}} \alpha(r-1) \langle \pi(r), \mathbf{d}(r-1) \rangle \right) \mathbf{1}, \end{aligned} \quad (11)$$

where $\langle \cdot, \cdot \rangle$ is used to denote the inner product of two vectors of proper dimension. Let $\mathbf{y}(\bar{t})$ denote the limiting vector of $\bar{\mathbf{x}}(\bar{t} + s + 1)$ as $s + 1 \rightarrow \infty$. Since all entries in the limiting vector are identical we denote the identical value by $y(\bar{t})$. Thus, $\mathbf{y}(\bar{t}) = [y(\bar{t}), \dots, y(\bar{t})]'$.

From (11) we have

$$y(\bar{t}) = \langle \pi(0), \mathbf{x}(0) \rangle - \sum_{r=1}^{\bar{t}} \alpha(r-1) \langle \pi(r), \mathbf{d}(r-1) \rangle. \quad (12)$$

If, instead, all agents cease computing $d_i(t)$ after iteration $\bar{t} + 1$, then the identical value, denoted by $y(\bar{t} + 1)$, equals

$$\begin{aligned} y(\bar{t} + 1) &= \langle \pi(0), \mathbf{x}(0) \rangle - \sum_{r=1}^{\bar{t}+1} \alpha(r-1) \langle \pi(r), \mathbf{d}(r-1) \rangle \\ &= \langle \pi(0), \mathbf{x}(0) \rangle - \sum_{r=1}^{\bar{t}} \alpha(r-1) \langle \pi(r), \mathbf{d}(r-1) \rangle - \alpha(\bar{t}) \langle \pi(\bar{t} + 1), \mathbf{d}(\bar{t}) \rangle \\ &= y(\bar{t}) - \alpha(\bar{t}) \langle \pi(\bar{t} + 1), \mathbf{d}(\bar{t}) \rangle, \end{aligned} \quad (13)$$

where each entry $d_i(\bar{t})$ in $\mathbf{d}(\bar{t})$ denotes the subgradient of $g_i(\cdot)$ computed by agent i at $x_i(\bar{t})$. With a little abuse of notation, henceforth we use t to replace \bar{t} . The actual reference of t should be clear from the context.

In our convergence analysis, we will use the well-know ‘almost supermartingale’ convergence theorem in [22], which can also be found as Lemma 11, in Chapter 2.2 [20]. We present a simpler deterministic version of the theorem in the next lemma.

Lemma 4. [22] *Let $\{a_t\}_{t=0}^\infty, \{b_t\}_{t=0}^\infty$, and $\{c_t\}_{t=0}^\infty$ be non-negative sequences. Suppose that*

$$a_{t+1} \leq a_t - b_t + c_t \quad \text{for all } t \geq 0,$$

and $\sum_{t=0}^\infty c_t < \infty$. Then $\sum_{t=0}^\infty b_t < \infty$ and the sequence $\{a_t\}_{t=0}^\infty$ converges to a non-negative value.

The basic iterative relation of the consensus value $y(t)$ is stated in our Lemma 5.

Lemma 5. *Let $\{y(t)\}_{t=0}^{\infty}$ be the sequence of limiting consensus value defined by (12), and $\{x_i(t)\}_{t=0}^{\infty}$ be the sequence for $i \in \mathcal{V} - \mathcal{F}$ generated by (7). Let $\{\delta_i(t)\}_{t=0}^{\infty}$ be a sequence of subgradients of g_i at $y(t)$ for all $i \in \mathcal{V} - \mathcal{F}$. Then the following basic relations hold. For any $x \in \mathbb{R}$ and any $t \geq 0$,*

$$\begin{aligned} |y(t+1) - x|^2 &\leq |y(t) - x|^2 + 4L\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| \\ &\quad - 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j(x)) + \alpha^2(t)(n-\phi)L^2 \end{aligned}$$

The proof of Lemma 5 can be found in [18]. We present the proof in Appendix E. For each t and each $i \in \mathcal{V} - \mathcal{F}$, the distance between the consensus value $y(t)$ and the local estimate $x_i(t)$ is bounded from above.

Lemma 6. *Let $U = \max_{i \in \mathcal{V} - \mathcal{F}} x_i(0)$, and $u = \min_{i \in \mathcal{V} - \mathcal{F}} x_i(0)$. For every $i \in \mathcal{V} - \mathcal{F}$, a uniform bound on $|y(t) - x_i(t)|$ for $t \geq 1$ is given by:*

$$|y(t) - x_i(t)| \leq (n-\phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{t}{\nu} \rceil} + (n-\phi) L \sum_{r=1}^{t-1} \alpha(r-1) \gamma^{\lceil \frac{t-r}{\nu} \rceil} + 2\alpha(t-1)L. \quad (14)$$

When $t = 1$, $\sum_{r=1}^{t-1} \alpha(r-1) \gamma^{\lceil \frac{t-r}{\nu} \rceil} = 0$ by convention.

Note that the upper bound on $|y(t) - x_i(t)|$ in (14) depends on t . In fact, this upper bound will diminish over time, as formally stated below.

Lemma 7. *For each $i \in \mathcal{V} - \mathcal{F}$, the limit of $|y(t) - x_i(t)|$ exists and*

$$\lim_{t \rightarrow \infty} |y(t) - x_i(t)| = 0.$$

Our main convergence result is stated below.

Theorem 4 (Convergence). *For each $i \in \mathcal{V} - \mathcal{F}$, $\{x_i(t)\}_{t=0}^{\infty}$ converges to the same optimum in X , i.e.,*

$$\lim_{t \rightarrow \infty} |x_i(t) - x^*| = 0,$$

where $x^* \in X$.

We provide a sketch of the convergence proof below. Formal proof can be found in Appendix E.

Recall that each $g_i(\cdot)$ is defined as

$$g_i(x) = \mathbf{A}_{1i}h_1(x) + \mathbf{A}_{2i}h_2(x) + \dots + \mathbf{A}_{ki}h_k(x),$$

for $i \in \mathcal{V}$, where $\mathbf{A}_{ji} \geq 0$ and $\sum_{j=1}^k \mathbf{A}_{ji} = 1$. Let $Y^i = \operatorname{argmin} g_i(x)$ and $Y_j^i = \operatorname{argmin} \mathbf{A}_{ji}h_j(x)$ for $j = 1, \dots, k$. Since for each $j \in \{1, \dots, k\}$ such that $\mathbf{A}_{ji} = 0$, $\operatorname{argmin} \mathbf{A}_{ji}h_j(x) = 0$ is a constant function over the whole real line, it holds that $Y_j^i = \mathbb{R}$. Since positive constant scaling does not affect the optimal set of a function, for each $j \in \{1, \dots, k\}$ such that $\mathbf{A}_{ji} > 0$, it holds that $Y_j^i = X_j$. In addition, because $h_1(x), \dots, h_k(x)$ are solution redundant functions, i.e., $\cap_{j=1}^k X_j \neq \emptyset$, functions

$\mathbf{A}_{1i}h_1(x), \dots, \mathbf{A}_{ki}h_k(x)$ are also solution redundant. It can be shown (formally proved in Appendix A) that

$$Y^i = \cap_{j:\mathbf{A}_{ji}>0} X_j \supseteq \cap_{j=1}^k X_j = X, \text{ for all } i \in \mathcal{V}.$$

Let $x' \in X$. Define g_j^* as the optimal value of function $g_j(\cdot)$ for each $j \in \mathcal{V}$. We have

$$\begin{aligned} |y(t+1) - x'|^2 &\leq |y(t) - x'|^2 + 4L\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| \\ &\quad - 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j(x')) + \alpha^2(t)(n-\phi)L^2 \\ &\stackrel{(a)}{=} |y(t) - x'|^2 + 4L\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| \\ &\quad - 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j^*) + \alpha^2(t)(n-\phi)L^2. \end{aligned} \quad (15)$$

Equality (a) holds because of $x' \in X \subseteq Y^j$ for each $j \in \mathcal{V}$, then $g_j(x') = g_j^*$.

For each $t \geq 0$, define

$$\begin{aligned} a_t &= |y(t) - x'|^2, \\ b_t &= 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j^*), \\ c_t &= 4L\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| + \alpha^2(t)(n-\phi)L^2. \end{aligned}$$

It is easy to see that $a_t \geq 0$ and $c_t \geq 0$ for each t . Since g_j^* is the optimal value of function $g_j(\cdot)$, it holds that $b_t \geq 0$ for each t . Thus, $\{a_t\}_{t=0}^\infty, \{b_t\}_{t=0}^\infty$ and $\{c_t\}_{t=0}^\infty$ are three non-negative sequences. By (15), it holds that

$$a_{t+1} \leq a_t - b_t + c_t \quad \text{for each } t \geq 0.$$

To apply Lemma 4, we need to show that $\sum_{t=0}^\infty c_t < \infty$. In fact, the following lemma holds.

Lemma 8.

$$\sum_{t=0}^\infty \alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| < \infty.$$

The proof of Lemma 8 is presented in Appendix E. In addition, since $\sum_{t=0}^\infty \alpha^2(t) < \infty$, it holds that

$$(n-\phi)L^2 \sum_{t=0}^\infty \alpha^2(t) < \infty.$$

Thus, we get

$$\begin{aligned}
\sum_{t=0}^{\infty} c_t &= \sum_{t=0}^{\infty} \left(4L\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| + \alpha^2(t)(n-\phi)L^2 \right) \\
&= 4L \sum_{t=0}^{\infty} \left(\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| \right) + (n-\phi)L^2 \sum_{t=0}^{\infty} \alpha^2(t) \\
&< \infty.
\end{aligned}$$

Therefore, applying Lemma 4 to the sequences $\{a_t\}_{t=0}^{\infty}$, $\{b_t\}_{t=0}^{\infty}$ and $\{c_t\}_{t=0}^{\infty}$, we have that for any $x' \in X$, $a_t = |y(t) - x'|$ converges, and

$$\sum_{t=0}^{\infty} b_t = \sum_{t=0}^{\infty} \alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j^*) < \infty.$$

Since $|y(t) - x'|$ converges for any fixed $x' \in X$, by definition of sequence convergence and the dynamic of $y(t)$ in (12), it is easy to see that $y(t)$ also converges. Let $\lim_{t \rightarrow \infty} y(t) = y$. Next we show that $y \in X$.

Let $\mathcal{I}_{t+1} \subseteq \mathcal{V} - \mathcal{F}$ be the set of indices such that for each $j \in \mathcal{I}_{t+1}$, $\pi_j(t+1) \geq \beta^\nu$. As $G(\mathcal{V}, \mathcal{E})$ satisfies Condition 1, $|\mathcal{I}_{t+1}| \geq \max\{k', f+1\}$. Since $g_j(y(t)) - g_j^* \geq 0$ for all j , then

$$\begin{aligned}
\sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j^*) &\geq \sum_{j \in \mathcal{I}_{t+1}} \pi_j(t+1) (g_j(y(t)) - g_j^*) \\
&\geq \beta^\nu \sum_{j \in \mathcal{I}_{t+1}} (g_j(y(t)) - g_j^*) \\
&= \beta^\nu \sum_{j \in \mathcal{I}_{t+1}} \sum_{i=1}^k \mathbf{A}_{ij} (h_i(y(t)) - h_i^*) \\
&= \beta^\nu \sum_{i=1}^k \left(\sum_{j \in \mathcal{I}_{t+1}} \mathbf{A}_{ij} \right) (h_i(y(t)) - h_i^*) \\
&\geq k\beta^\nu C_2 (h(y(t)) - h^*),
\end{aligned}$$

where

$$C_2 = \min_{\mathcal{I} \subseteq \mathcal{V}: |\mathcal{I}| \geq \max\{k', f+1\}} \sum_{i \in \mathcal{I}} \mathbf{A}_{ij},$$

and the last inequality follows from the fact that $h_i(y(t)) - h_i^* \geq 0$. In addition, as $sp(\mathbf{A}) = k'$, then $\sum_{i \in \mathcal{I}} \mathbf{A}_{ij} > 0$ for every $\mathcal{I} \subseteq \mathcal{V} : |\mathcal{I}| \geq \max\{k', f+1\}$. Since \mathbf{A} is finite, C_2 is well-defined and $C_2 > 0$. If $y \notin X$, it can be shown that $k\beta^\nu C_2 (h(y(t)) - h^*) = \infty$. This contradicts the fact that $\sum_{t=0}^{\infty} b_t < \infty$. Thus, $y \in X$.

Therefore, we conclude that limit of $|x_i(t) - y|$ exists and

$$\lim_{t \rightarrow \infty} |x_i(t) - y| = 0,$$

proving Theorem 4.

5 Summary and Conclusion

In this report, we introduce the condition-based approach to Byzantine multi-agent optimization. We have shown that when there is enough redundancy in the local cost functions, or in the local optima, Problem 1 can be solved iteratively.

Two slightly different variants are considered: condition-based Byzantine multi-agent optimization with side information and condition-based Byzantine multi-agent optimization without side information. For the former, when side information is available at each agent, a decoding-based algorithm is proposed, assuming each input function is differentiable. This algorithm combines the gradient method with the decoding procedure introduced in [4] by choosing proper “generator matrices” as job assignment matrices. With such a decoding subroutine, our algorithm essentially performs the gradient method, where gradient computation is processed distributedly over the multi-agent system. When side information is not available at each agent, we propose a simple consensus-based algorithm in which each agent carries minimal state across iterations. This consensus-based algorithm solves Problem 1 under the additional assumption over input functions that all input functions share at least one common optimum. Although the consensus-based algorithm can only solve Problem 1 for a restricted class of input functions, nevertheless, as each non-faulty agent does not need to store the job matrix \mathbf{A} throughout execution and does not need to perform the decoding procedure at each iteration, the requirements on memory and computation are less stringent comparing to the decoding-based algorithm. In addition, in contrast to the decoding-based algorithm, the consensus-based algorithm also works for nonsmooth input functions. Thus, the consensus-based algorithm may be more practical in some applications.

References

1. J. Anthonisse and H. Tijms. Exponential convergence of products of stochastic matrices. *Journal of Mathematical Analysis and Applications*, 59(2):360 – 364, 1977.
2. D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., 1989.
3. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan. 2011.
4. E. J. Candes and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.
5. S. Chatterjee and E. Seneta. Towards consensus: Some convergence theorems on repeated averaging. *Journal of Applied Probability*, 14(1):pp. 89–97, 1977.
6. S. Chaudhuri. More choices allow more faults: Set consensus problems in totally asynchronous systems. *Information and Computation*, 105:132–158, 1992.
7. D. Dolev, N. A. Lynch, S. S. Pinter, E. W. Stark, and W. E. Weihl. Reaching approximate agreement in the presence of faults. *J. ACM*, 33(3):499–516, May 1986.
8. J. Duchi, A. Agarwal, and M. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *Automatic Control, IEEE Transactions on*, 57(3):592–606, March 2012.
9. A. D. Fekete. Asymptotically optimal algorithms for approximate agreement. *Distributed Computing*, 4(1):9–29, 1990.
10. R. Friedman, A. Mostefaoui, S. Rajsbaum, and M. Raynal. Asynchronous agreement and its relation with error-correcting codes. *Computers, IEEE Transactions on*, 56(7):865–875, 2007.
11. B. Kailkhura, S. Brahma, and P. K. Varshney. Consensus based detection in the presence of data falsification attacks. *arXiv preprint arXiv:1504.03413*, 2015.
12. H. J. LeBlanc, H. Zhang, S. Sundaram, and X. Koutsoukos. Consensus of multi-agent networks in the presence of adversaries using only local information. In *Proceedings of the 1st International Conference on High Confidence Networked Systems*, HiCoNS ’12, pages 1–10, New York, NY, USA, 2012. ACM.
13. S. Marano, V. Matta, and L. Tong. Distributed detection in the presence of byzantine attacks. *Signal Processing, IEEE Transactions on*, 57(1):16–29, 2009.
14. A. Mostefaoui, S. Rajsbaum, and M. Raynal. Conditions on input vectors for consensus solvability in asynchronous distributed systems. *Journal of the ACM (JACM)*, 50(6):922–954, 2003.

15. A. Mostefaoui, S. Rajsbaum, and M. Raynal. Using conditions to expedite consensus in synchronous distributed systems. In *Distributed Computing*, pages 249–263. Springer, 2003.
16. A. Mostefaoui, S. Rajsbaum, and M. Raynal. Synchronous condition-based consensus. *Distributed Computing*, 18(5):325–343, 2006.
17. A. Nedic and A. Olshevsky. Distributed optimization over time-varying directed graphs. *Automatic Control, IEEE Transactions on*, 60(3):601–615, 2015.
18. A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *Automatic Control, IEEE Transactions on*, 54(1):48–61, Jan 2009.
19. M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, Apr. 1980.
20. B. T. Poljak. *Introduction to optimization*. Optimization Software, 1987.
21. S. S. Ram, A. Nedić, and V. V. Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of optimization theory and applications*, 147(3):516–545, 2010.
22. H. Robbins and D. Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In T. Lai and D. Siegmund, editors, *Herbert Robbins Selected Papers*, pages 111–135. Springer New York, 1985.
23. L. Su and N. Vaidya. Byzantine multi-agent optimization: Part I. *arXiv preprint arXiv:1506.04681*, 2015.
24. K. I. Tsianos, S. Lawlor, and M. G. Rabbat. Push-sum distributed dual averaging for convex optimization. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 5453–5458, Dec 2012.
25. J. N. Tsitsiklis, D. P. Bertsekas, M. Athans, et al. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.
26. N. H. Vaidya. Matrix representation of iterative approximate byzantine consensus in directed graphs. *CoRR*, abs/1203.1888, 2012.
27. N. H. Vaidya, L. Tseng, and G. Liang. Iterative approximate byzantine consensus in arbitrary directed graphs. In *Proceedings of the 2012 ACM symposium on Principles of distributed computing*, pages 365–374. ACM, 2012.
28. N. H. Vaidya, L. Tseng, and G. Liang. Iterative approximate byzantine consensus in arbitrary directed graphs. In *Proceedings of the 2012 ACM Symposium on Principles of Distributed Computing*, PODC ’12, pages 365–374, New York, NY, USA, 2012. ACM.
29. P. Zhang, J. Y. Koh, S. Lin, and I. Nevat. Distributed event detection under byzantine attack in wireless sensor networks. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6. IEEE, 2014.

Appendices

A Connection between X and X_j 's for $j = 1, \dots, k$

Recall that X_j is the set of optimal solution(s) of input function $h_j(x)$, for $j = 1, \dots, k$; and that X is the optimal set of function $\frac{1}{k} \sum_{j=1}^k h_j(x)$ in Problem 1. Propositions 1 and 2 are used in proving the correctness of other results in this report.

Proposition 1. [23] *The optimal set X of Problem 1 is contained in the convex hull of the union of all X_j 's, i.e.,*

$$X \subseteq \text{Cov} \left(\bigcup_{j=1}^k X_j \right), \quad (16)$$

where $\text{Cov}(Z)$ is the convex hull of set Z .

The above proposition holds for any collection of k admissible input functions. A stronger connection holds for solution-redundant input functions, as stated below.

Proposition 2. *When all the input functions share at least one common optimum, i.e., $\bigcap_{j=1}^k X_j \neq \emptyset$, then*

$$X = \bigcap_{j=1}^k X_j. \quad (17)$$

Proof. We first show that $\bigcap_{j=1}^k X_j \subseteq X$. Let h_j^* be the optimal value of function $h_j(x)$ for $j = 1, \dots, k$, and let h^* be the optimal value of function $\frac{1}{k} \sum_{j=1}^k h_j(x)$. Since $\bigcap_{j=1}^k X_j \neq \emptyset$, let $x_0 \in \bigcap_{j=1}^k X_j$. Then for all $x \in \mathbb{R}$

$$\frac{1}{k} \sum_{j=1}^k h_j(x_0) = \frac{1}{k} \sum_{j=1}^k h_j^* \leq h^* \leq \frac{1}{k} \sum_{j=1}^k h_j(x).$$

So we know

$$\sum_{j=1}^k h_j(x_0) = \sum_{j=1}^k h_j^* = h^*.$$

Thus $x_0 \in X$ and $\bigcap_{j=1}^k X_j \subseteq X$.

Next we show $X \subseteq \bigcap_{j=1}^k X_j$. We prove this by contradiction. Suppose on the contrary that $X \not\subseteq \bigcap_{j=1}^k X_j$, then there exists $x' \in X$ such that $x' \notin \bigcap_{j=1}^k X_j$. The latter implies that $x' \notin X_{j_0}$ for

some $j_0 \in \{1, \dots, k\}$. Then

$$\begin{aligned}
\sum_{j=1}^k h_j(x') &= \left(\sum_{1 \leq j \leq k, j \neq j_0} h_j(x') \right) + h_{j_0}(x') \\
&\geq \left(\sum_{1 \leq j \leq k, j \neq j_0} h_j^* \right) + h_{j_0}(x') \\
&> \left(\sum_{1 \leq j \leq k, j \neq j_0} h_j^* \right) + h_{j_0}^* \\
&= \sum_{j=1}^k h_j^* = h^*.
\end{aligned}$$

So $x' \notin X$, which leads to a contradiction. Thus $X \subseteq \cap_{j=1}^k X_j$.

Therefore, $X = \cap_{j=1}^k X_j$.

□

B Condition-based Byzantine multi-agent optimization without side information

Proof of Lemma 1

Proof. Let $sp(\mathbf{A}) = k'$, by definition of $sp(\mathbf{A})$, the sum vector of any collection of k' columns of \mathbf{A} is component-wise positive. Suppose there exists a row, say i_0 , that contains at least k' zero entries. Let $j_1, j_2, \dots, j_{k'}$ be any k' columns in \mathbf{A} , wherein the i_0 -coordinate of each column is zero. Thus the i_0 -th coordinate of $\sum_{r=1}^{k'} \mathbf{A}_{j_r}$ is zero, contradicting the hypothesis that $sp(\mathbf{A}) = k'$. In addition, if every row contains more than $k' - 1$ zeros, using the same argument it can be shown that k' is not the smallest integer.

Conversely, we need to show that if there are at most $k' - 1$ zero entries in each row of \mathbf{A} and there exists one row contains exactly $k' - 1$ zero entries, then $sp(\mathbf{A}) = k'$. Let $j_1, \dots, j_{k'}$ be any collection of k' columns of \mathbf{A} . For each coordinate, at least one of the chosen k' columns contains positive entry in that coordinate. So we have $\sum_{r=1}^{k'} \mathbf{A}_{j_r} > \mathbf{0}$ componentwise. By definition of $sp(\mathbf{A})$, we know $sp(\mathbf{A}) \leq k'$. In addition, let i_0 be a row in which there are exactly $k' - 1$ zeros, then there exists a collection of $k' - 1$ columns whose i_0 -th coordinate are all zeros, and that the sum of the $k' - 1$ columns also has the i_0 -th coordinate being 0. Thus $sp(\mathbf{A}) = k'$.

□

Proof of Theorem 2

Proof. We first show that if Problem 1 is solvable, then a source component must exist in every reduced graph of $G(\mathcal{V}, \mathcal{E})$, containing at least $f + 1$ nodes. Then we show that when $k' > f + 1$, the source component must contain at least k' nodes. These two claims together show that if Problem 1 is solvable, then a source component must exist in every reduced graph of $G(\mathcal{V}, \mathcal{E})$, containing at least $\max\{f + 1, k'\}$ nodes, proving the theorem.

Definition 4 (Condition 2). *Given a graph $G(\mathcal{V}, \mathcal{E})$, for any node partition L, R, C, F of $G(\mathcal{V}, \mathcal{E})$ such that L, R are nonempty and $|F| \leq f$, one of the following must hold: (1) there exists a node $i \in L$ that has at least $f + 1$ incoming neighbors in $R \cup C$, i.e., $|N_i^- \cap (R \cup C)| \geq f + 1$; or (2) there exists a node $j \in R$ that has at least $f + 1$ incoming neighbors in $L \cup C$, i.e., $|N_j^- \cap (L \cup C)| \geq f + 1$.*

Now we show that Condition 2 is a necessary condition for Problem 1. Suppose graph $G(\mathcal{V}, \mathcal{E})$ does not satisfy Condition 2 and there exists a correct algorithm \mathcal{A} solving Problem 1 for solution redundant input functions. Recall that $X_i = \operatorname{argmin} h_i(x)$ for each $i = 1, \dots, k$, and $X = \operatorname{argmin} h(x)$. Consider the input functions $h_i(x)$ for all $i = 1, \dots, k$ such that each $h_i(x)$ is admissible with optimal set $X_i = [0, 1]$. Since $\bigcap_{i=1}^k X_i = [0, 1] \neq \emptyset$, then $h_1(x), \dots, h_k(x)$ is a collection of k solution redundant input functions. In addition, by Proposition 2, we know that $X = [0, 1]$. Since $G(\mathcal{V}, \mathcal{E})$ does not satisfy Condition 2, then there exists a node partition L, R, C, F , where L, R are nonempty and $|F| \leq f$, such that $|N_i^- \cap (R \cup C)| \leq f$ for each $i \in L$ and $|N_j^- \cap (L \cup C)| \leq f$ for each $j \in R$. Consider the execution, denoted by e_1 , wherein all nodes in F are faulty and all the remaining nodes are non-faulty. The initial states of all non-faulty nodes are assigned as follows: $x_i(0) = 0$ for each $i \in L$, $x_i(0) = 1$ for each $i \in R$, and $x_i(0)$ as an arbitrary value within $[0, 1]$ for each $i \in C$. In iteration 1, each faulty node p sends $F_p(0, g_p(\cdot))$ to nodes in L , sends $F_p(1, g_p(\cdot))$ to nodes in R and sends $F_p(a, g_p(\cdot))$ to nodes in C , where a is an arbitrary value within $[0, 1]$. Let $i \in L$ be an arbitrary node in L . We will show that there exists an execution e_i that can not be distinguished from e_1 by node i . Thus node i should behave in the same way in e_1 and e_i . Let $x_i(t)$ and $\bar{x}_i(t)$ be the local estimate of agent i in e_1 and in e_i , respectively.

Execution e_i : The input functions are $h_1(x), \dots, h_k(x)$. All nodes in $N_i^- \cap (R \cup C)$ are faulty, and the other nodes are non-faulty with initial state 0, i.e., $\bar{x}_j(0) = 0$ for all $j \notin N_i^- \cap (R \cup C)$.⁴ Since $\bar{x}_i(0) = 0 \in [0, 1] = X$, for all $i \in \mathcal{V} - \mathcal{F}$, then $\bar{x}_i(1) = 0$ for all $i \in \mathcal{V} - \mathcal{F}$, where $\mathcal{F} = N_i^- \cap (R \cup C)$ is the set of faulty nodes in execution e_i .

Since agent i cannot distinguish execution e_i from e_1 , thus $x_i(1) = 0$. As agent i is an arbitrary agent in L , in execution e_1 it holds that $x_i(1) = 0$ for all $i \in L$. Similarly, we can show that in execution e_1 , $x_i(1) = 1$ for all $i \in R$. Repeatedly applying the above argument, we can conclude that for any iteration t in execution e_1 , it follows that $x_i(t) = 0$ for all $i \in L$ and $x_j(t) = 1$ for all $j \in R$, contradicting the asymptotic consensus requirement of a correct algorithm for Problem 1. Thus, Condition 2 is a necessary condition for Problem 1. In addition, it was shown in [28] that graph $G(\mathcal{V}, \mathcal{E})$ satisfies Condition 2 if and only if a source component exists containing at least $f + 1$ nodes in every reduced graph of $G(\mathcal{V}, \mathcal{E})$.

Therefore, if Problem 1 is solvable, a source component containing at least $f + 1$ nodes must exist in every reduced graph of $G(\mathcal{V}, \mathcal{E})$. Next we show, by contradiction, that if Problem 1 is solvable and $k' > f + 1$, a source component must contain at least k' nodes.

Suppose there exists a reduced graph \mathcal{H} of $G(\mathcal{V}, \mathcal{E})$ whose source component contains at most $k' - 1$ agents, and there exists a correct algorithm \mathcal{A} that can solve Problem 1. Denote the source component of the reduced graph \mathcal{H} by $S_{\mathcal{H}}$. Let L, R, C, F be the node partition of $G(\mathcal{V}, \mathcal{E})$ where $L = S_{\mathcal{H}}$, $C = \emptyset$, $R = \mathcal{V} - L - \mathcal{F}$ and $F = \mathcal{F}$. Note that it is possible that $R = \emptyset$. Let $h_1(x), \dots, h_k(x)$ and $\tilde{h}_1(x), \dots, \tilde{h}_k(x)$ be two collections of k admissible input functions such that (1) $h_i(\cdot) = \tilde{h}_i(\cdot)$ for $i = 1, \dots, k - 1$, $\operatorname{argmin} \tilde{h}_i(x) = [0, 1]$ for all $i = 1, \dots, k$, and $\operatorname{argmin} h_k(x) = \{1\}$. Let \mathbf{A} be an assignment matrix such that $sp(\mathbf{A}) = k'$ and $\mathbf{A}_{ki} = 0$ for each $i \in L$. Such a matrix exists, since $|L| \leq k' - 1$. Informally speaking, with this assignment matrix, each agent i in L does not

⁴ Execution e_i is possible since $|N_i^- \cap (R \cup C)| \leq f$.

have any information about the k -th input function. Consider the execution E_1 , wherein the input functions are $h_1(x), \dots, h_k(x)$, each agent p in F is faulty and all other agents are non-faulty. The initial states of non-faulty agents in execution E_1 are assigned as follows: $x_i(0) = 0$ for all $i \in L$, and $x_i(0) = 1$ for all $i \in R$ (if $R \neq \emptyset$)—recalling that $C = \emptyset$. Each faulty agent $p \in F$ sends $F_p \left(0, \sum_{i=1}^k \mathbf{A}_{ip} \tilde{h}_i(x)\right)$ to nodes in L , and sends $F_p \left(1, \sum_{i=1}^k \mathbf{A}_{ip} h_i(x)\right)$ to nodes in R (if $R \neq \emptyset$).

Let $i \in L$ be an arbitrary agent in L . Now consider the following execution, denoted by E_i , wherein the local estimate of each non-faulty node j is denoted as \bar{x}_j . The input functions are $\tilde{h}_1(x), \dots, \tilde{h}_k(x)$. All nodes in $N_i^- \cap (R \cup C)$ are faulty, and the other nodes are non-faulty with initial state 0, i.e., $\bar{x}_j(0) = 0$ for all $j \notin N_i^- \cap (R \cup C)$. Since $\bar{x}_i(0) = 0 \in [0, 1] = X$, for all $i \in \mathcal{V} - \mathcal{F}$, then $\bar{x}_i(t) = 0$ for all $i \in \mathcal{V} - \mathcal{F}$ and for all t , where $\mathcal{F} = N_i^- \cap (R \cup C)$ is the set of faulty nodes in execution E_i .

Node i cannot distinguish E_i from E_1 , thus $x_i(1) = 0$ in E_1 . Since i is an arbitrary node in L , thus $x_i(1) = 0$ for all $i \in L$ in E_1 . Repeatedly applying the above argument, we have $\lim_{t \rightarrow \infty} x_i(t) = 0$ for each $i \in L$ in E_1 . However, we know that in E_1 , the optimal set is $X = \{1\}$. Because in execution E_1 , the correct output must be 1, \mathcal{A} is not a correct algorithm. Thus, we know if Problem 1 is solvable and $k' > f + 1$, a source component must contain at least k' nodes.

Therefore, we conclude that if Problem 1 is solvable, a source component containing at least $\max\{f + 1, k'\}$ nodes must exist in every reduced graph of $G(\mathcal{V}, \mathcal{E})$, proving Theorem 2. \square

Proof of Corollary 1

Proof. Let $sp(\mathbf{A}) = k'$. It was shown in [27] that if Condition 2 is true, then $n \geq 3f + 1$. It is enough to consider the case when $k' > f + 1$.

Suppose $3f + 1 \leq n < k' + 2f$. Consider the node partition L, R, F such that $|R| = |F| = f$, and $L = \mathcal{V} - R - F$. Since $3f + 1 \leq n < k' + 2f$, it holds that $f + 1 \leq |L| \leq k' - 1$. Suppose all nodes in F are faulty. Consider the subgraph \mathcal{H} constructed from $G(\mathcal{V}, \mathcal{E})$ by (1) removing all faulty nodes, i.e., all nodes in F , and (2) for each $i \in L$, removing all incoming links from R . The subgraph \mathcal{H} is a valid reduced graph since $|R| = f$. By Theorem 2, a source component exists in \mathcal{H} . Let S be the source component of \mathcal{H} . By Theorem 2, it holds that $|S| \geq k'$. Since each node $j \in R$ cannot reach nodes in L , by definition, each node $j \in R$ is not contained in a source component. Thus, $S \subseteq L$. Consequently, it holds that $|S| \leq |L| \leq k' - 1$, contradicting the fact that $|S| \geq k'$. Thus, when $k' > f + 1$, it holds that $n \geq k' + 2f$.

Therefore, we conclude that $n \geq \max\{3f + 1, k' + 2f\}$. \square

C Matrix Representation of Algorithm 2

If $G(\mathcal{V}, \mathcal{E})$ satisfies Condition 1, it was shown in Proposition 3 that the updates of $\mathbf{x} \in \mathbb{R}^{n-\phi}$ in each iteration can be written compactly in a matrix form. This observation is made in [26], and we restate this result below for completeness.

Proposition 3. [26] We can express the iterative update of the state of a non-faulty node i ($1 \leq i \leq n - \phi$) performed in (4) using the matrix form in (18) below, where $\mathbf{M}_i(t)$ satisfies the following four conditions.

$$x_i(t+1) = \mathbf{M}_i(t) \mathbf{x}(t) - \alpha(t)d_i(t). \quad (18)$$

In addition to t , the row vector $\mathbf{M}_i(t)$ may depend on the state vector $\mathbf{x}(t-1)$ as well as the behavior of the faulty nodes in \mathcal{F} . For simplicity, the notation $\mathbf{M}_i(t)$ does not explicitly represent this dependence.

1. $\mathbf{M}_i(t)$ is a stochastic row vector of size $(n - \phi)$. Thus, $\mathbf{M}_{ij}(t) \geq 0$, for $1 \leq j \leq n - \phi$, and

$$\sum_{1 \leq j \leq n - \phi} \mathbf{M}_{ij}(t) = 1$$

2. $\mathbf{M}_{ii}(t)$ equals a_i defined in Algorithm 1. Recall that $a_i \geq \alpha$.
3. $\mathbf{M}_{ij}(t)$ is non-zero **only if** $(j, i) \in \mathcal{E}$ or $j = i$.
4. At least $|N_i^- \cap (\mathcal{V} - \mathcal{F})| - f + 1$ elements in $\mathbf{M}_i(t)$ are lower bounded by some constant $\beta > 0$ (β is independent of i). Note that $N_i^- \cap (\mathcal{V} - \mathcal{F})$ is the set of non-faulty incoming neighbors of node i .

D Convergence of the Transition Matrices $\Phi(t, r)$

Proof of Lemma 2

Proof. Recall that $R_{\mathcal{F}}$ is the collection of all reduced graphs of the given graph $G(\mathcal{V}, \mathcal{E})$. Let $\mathcal{H} \in R_{\mathcal{F}}$ be an arbitrary reduced graph with adjacency matrix \mathbf{H} . Let $k' = sp(\mathbf{A})$. From Theorem 2 we know that there are at least $\max\{k', f + 1\}$ nodes in the unique source component in \mathcal{H} . Denote the source component in \mathcal{H} by $S_{\mathcal{H}}$ and let j_1, j_2, \dots, j_p , where

$$p \triangleq |S_{\mathcal{H}}| \geq \max\{k', f + 1\},$$

be the p nodes in $S_{\mathcal{H}}$. By definition, each j_i has a directed path to all the other non-faulty nodes in \mathcal{H} . Since the length of a path from j_i to any other node in \mathcal{H} is at most $n - \phi - 1$, then the j_i -th column of $\mathbf{H}^{n-\phi}$ will be non-zero for $i = 1, 2, \dots, p$. Since $p \geq \max\{k', f + 1\}$, there are at least $\max\{k', f + 1\}$ such columns in $\mathbf{H}^{n-\phi}$.

Recall that for any $t \geq 1$, there exists a graph $\mathcal{H}(t) \in R_{\mathcal{F}}$ such that $\beta \mathbf{H}(t) \leq \mathbf{M}(t)$, thus we have

$$\begin{aligned} \Phi(r + \nu - 1, r) &= \mathbf{M}(r + \nu - 1) \mathbf{M}(r + \nu - 2) \dots \mathbf{M}(r) \\ &\geq \beta^\nu \prod_{t=r}^{r+\nu-1} \mathbf{H}(t). \end{aligned}$$

The above product of adjacency matrices consists of $\nu = \tau(n - \phi)$ matrices (corresponding to reduced graphs) in $R_{\mathcal{F}}$. Thus, at least one of the τ distinct adjacency matrices in $R_{\mathcal{F}}$, say \mathcal{H}' , will appear in the above product at least $n - \phi$ times. Let $S_{\mathcal{H}'}$ and \mathbf{B} be the source component size and the adjacency matrix, respectively, of \mathcal{H}' . In addition, let $p' \triangleq |\mathcal{H}'|$. Due to the existence

of self-loops in the update dynamic, each $\mathbf{H}(t)$ has positive diagonal. In addition, $(\mathbf{B})^{n-\phi}$ contains p' nonzero columns, where $p' \geq \max\{k', f+1\}$. Thus each of the $j_1, j_2, \dots, j_{p'}$ columns in $\prod_{t=r}^{r+\nu-1} \mathbf{H}(t)$ is lowered by $\mathbf{1} \in \mathbb{R}^{n-\phi}$ component-wise, i.e., $\left(\prod_{t=r}^{r+\nu-1} \mathbf{H}(t)\right)_{\cdot j_i} \geq \mathbf{1}$ for $i = 1, \dots, p'$, where $\left(\prod_{t=r}^{r+\nu-1} \mathbf{H}(t)\right)_{\cdot j_i}$ is the j_i -th column of $\prod_{t=r}^{r+\nu-1} \mathbf{H}(t)$. Therefore,

$$(\Phi(r+\nu-1, r))_{\cdot j_i} \geq \beta^\nu \mathbf{1},$$

for $i = 1, 2, \dots, p'$, where $p' \geq \max\{k', f+1\}$ —noting that $(\Phi(r+\nu-1, r))_{\cdot j_i}$ is the j_i -th column of $\Phi(r+\nu-1, r)$. □

Proof of Lemma 3

Proof. From Lemma 2, we know that there are at least $\max\{sp(\mathbf{A}), f+1\}$ columns of $\Phi(r+\nu-1, r)$ that are lower bounded by β^ν for a given r . Let \mathcal{I}_r be the collection of column indices such that for each $i \in \mathcal{I}_r$,

$$(\Phi(r+\nu-1, r))_{\cdot i} \geq \beta^\nu \mathbf{1}.$$

Let $t > r + \nu$. From (8), we know that

$$\lim_{t \geq r, t \rightarrow \infty} \Phi(t, r) = \mathbf{1}\pi(r)',$$

for all r . By the definition of $\Phi(t, r)$, we know for $t \geq r + \nu - 1$, we have

$$\Phi(t, r) = \Phi(t, r + \nu) \Phi(r + \nu - 1, r).$$

Thus,

$$\begin{aligned} \mathbf{1}\pi(r)' &= \lim_{t \geq r, t \rightarrow \infty} \Phi(t, r) \\ &= \lim_{t \geq r, t \rightarrow \infty} \Phi(t, r + \nu) \Phi(r + \nu - 1, r) \\ &= (\mathbf{1}\pi(r + \nu)') \Phi(r + \nu - 1, r). \end{aligned}$$

Thus, for each $i \in \mathcal{I}_r$,

$$\begin{aligned} \pi_i(r) &= \sum_{j=1}^{n-\phi} \pi_j(r + \nu) \Phi_{ji}(r + \nu - 1, r) \\ &\geq \left(\sum_{j=1}^{n-\phi} \pi_j(r + \nu) \right) \beta^\nu \\ &\geq \beta^\nu. \end{aligned}$$

□

E Convergence Analysis of Algorithm 1

Proof of Lemma 5

The proof of Lemma 5 can be found in [18]. We present the proof below for completeness.

Proof. For any $x \in \mathbb{R}$ and any $t \geq 0$,

$$\begin{aligned}
|y(t+1) - x|^2 &= |y(t) - \alpha(t) \langle \pi(t+1), \mathbf{d}(t) \rangle - x|^2 \\
&= |y(t) - x|^2 - 2\alpha(t) \langle \pi(t+1), \mathbf{d}(t) \rangle (y(t) - x) + \alpha^2(t) |\langle \pi(t+1), \mathbf{d}(t) \rangle|^2 \\
&\stackrel{(a)}{\leq} |y(t) - x|^2 - 2\alpha(t) \langle \pi(t+1), \mathbf{d}(t) \rangle (y(t) - x) + \alpha^2(t) \|\pi(t+1)\|^2 \|\mathbf{d}(t)\|^2 \\
&\stackrel{(b)}{\leq} |y(t) - x|^2 - 2\alpha(t) \langle \pi(t+1), \mathbf{d}(t) \rangle (y(t) - x) + \alpha^2(t) \|\mathbf{d}(t)\|^2 \\
&= |y(t) - x|^2 - 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) d_j(t) (y(t) - x) + \alpha^2(t) \sum_{j=1}^{n-\phi} d_j^2(t).
\end{aligned}$$

Inequality (a) follows from Cauchy-Schwarz inequality. Inequality (b) follows because

$$\|\pi(t+1)\|^2 = \sum_{j=1}^{n-\phi} \pi_j^2(t+1) \leq \sum_{j=1}^{n-\phi} \pi_j(t+1) = 1.$$

We now consider the term $d_j(t) (y(t) - x)$ for any $j \in \mathcal{V} - \mathcal{F}$, for which we have

$$\begin{aligned}
d_j(t) (y(t) - x) &= d_j(t) (y(t) - x_j(t) + x_j(t) - x) \\
&= d_j(t) (y(t) - x_j(t)) + d_j(t) (x_j(t) - x) \\
&\geq -|d_j(t)| |y(t) - x_j(t)| + d_j(t) (x_j(t) - x) \\
&\geq -|d_j(t)| |y(t) - x_j(t)| + g_j(x_j(t)) - g_j(x),
\end{aligned} \tag{19}$$

since $d_j(t)$ is a gradient of $g_j(\cdot)$ at $x_j(t)$. Furthermore, by using a gradient $\delta_j(t)$ of $g_j(\cdot)$ at $y(t)$, we also have for any $j \in \mathcal{V} - \mathcal{F}$ and $x \in \mathbb{R}$,

$$\begin{aligned}
g_j(x_j(t)) - g_j(x) &= g_j(x_j(t)) - g_j(y(t)) + g_j(y(t)) - g_j(x) \\
&\geq \delta_j(t) (x_j(t) - y(t)) + g_j(y(t)) - g_j(x) \\
&\geq -|\delta_j(t)| |x_j(t) - y(t)| + g_j(y(t)) - g_j(x).
\end{aligned} \tag{20}$$

Combining (19) and (20) together, it follows that for any $j \in \mathcal{V} - \mathcal{F}$ and any $x \in \mathbb{R}$, we obtain

$$d_j(t) (y(t) - x) \geq -(|d_j(t)| + |\delta_j(t)|) |y(t) - x_j(t)| + g_j(y(t)) - g_j(x).$$

Therefore,

$$\begin{aligned}
|y(t+1) - x|^2 &\leq |y(t) - x|^2 - 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) d_j(t) (y(t) - x) + \alpha^2(t) \sum_{j=1}^{n-\phi} d_j^2(t) \\
&\leq |y(t) - x|^2 \\
&\quad + 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) ((|d_j(t)| + |\delta_j(t)|) |y(t) - x_j(t)| - (g_j(y(t)) - g_j(x))) \\
&\quad + \alpha^2(t) \sum_{j=1}^{n-\phi} d_j^2(t) \\
&\leq |y(t) - x|^2 + 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) ((|d_j(t)| + |\delta_j(t)|) |y(t) - x_j(t)|) \\
&\quad - 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j(x)) + \alpha^2(t) \sum_{j=1}^{n-\phi} d_j^2(t) \\
&\leq |y(t) - x|^2 + 4L\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| \\
&\quad - 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j(x)) + \alpha^2(t)(n-\phi)L^2.
\end{aligned}$$

The last inequality holds from the fact that $g_j(\cdot)$ is L -Lipschitz continuous for each $j \in \mathcal{V}$.

□

Proof of Lemma 6

Proof. Recall (7). For $t > 0$,

$$\mathbf{x}(t) = \Phi(t-1, 0)\mathbf{x}(0) - \sum_{r=1}^t \alpha(r-1)\Phi(t-1, r)\mathbf{d}(r-1)$$

then each $x_i(t)$ can be written as

$$x_i(t) = \sum_{j=1}^{n-\phi} \Phi_{ij}(t-1, 0)x_j(0) - \sum_{r=1}^t \left(\alpha(r-1) \sum_{j=1}^{n-\phi} \Phi_{ij}(t-1, r)d_j(r-1) \right);$$

and (12) implies that $y(t) = \sum_{j=1}^{n-\phi} \pi_j(0)x_j(0) - \sum_{r=1}^t \alpha(r-1) \sum_{j=1}^{n-\phi} \pi_j(r)d_j(r-1)$. Thus

$$\begin{aligned}
&|y(t) - x_i(t)| \\
&\leq \left| \sum_{j=1}^{n-\phi} (\pi_j(0) - \Phi_{ij}(t-1, 0)) x_j(0) \right| + \left| \sum_{r=1}^t \left(\alpha(r-1) \sum_{j=1}^{n-\phi} (\Phi_{ij}(t-1, r) - \pi_j(r)) d_j(r-1) \right) \right|.
\end{aligned} \tag{21}$$

We bound the two terms in (21) separately. For the first term in (21), we have

$$\begin{aligned}
\left| \sum_{j=1}^{n-\phi} (\pi_j(0) - \Phi_{ij}(t-1, 0)) x_j(0) \right| &\leq \sum_{j=1}^{n-\phi} |\pi_j(0) - \Phi_{ij}(t-1, 0)| |x_j(0)| \\
&\stackrel{(a)}{\leq} \sum_{j=1}^{n-\phi} \gamma^{\lceil \frac{t}{\nu} \rceil} \max\{|u|, |U|\} \\
&= (n-\phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{t}{\nu} \rceil},
\end{aligned} \tag{22}$$

where inequality (a) follows from Theorem 3.

In addition, the second term in (21) can be bounded as follows.

$$\begin{aligned}
&\left| \sum_{r=1}^t \left(\alpha(r-1) \sum_{j=1}^{n-\phi} (\Phi_{ij}(t-1, r) - \pi_j(r)) d_j(r-1) \right) \right| \\
&\stackrel{(a)}{\leq} \sum_{r=1}^{t-1} \left(\alpha(r-1) \sum_{j=1}^{n-\phi} |\Phi_{ij}(t-1, r) - \pi_j(r)| |d_j(r-1)| \right) + \alpha(t-1) \left| d_i(t-1) - \sum_{j=1}^{n-\phi} \pi_j(t) d_j(t-1) \right| \\
&\leq \sum_{r=1}^{t-1} \left(\alpha(r-1) \sum_{j=1}^{n-\phi} |\Phi_{ij}(t-1, r) - \pi_j(r)| |d_j(r-1)| \right) + \alpha(t-1) \sum_{j=1}^{n-\phi} \pi_j(t) |d_i(t-1) - d_j(t-1)| \\
&\leq \sum_{r=1}^{t-1} \left(\alpha(r-1) \sum_{j=1}^{n-\phi} |\Phi_{ij}(t-1, r) - \pi_j(r)| \right) L + 2\alpha(t-1)L \\
&\leq (n-\phi) L \sum_{r=1}^{t-1} \alpha(r-1) \gamma^{\lceil \frac{t-r}{\nu} \rceil} + 2\alpha(t-1)L
\end{aligned} \tag{23}$$

where inequality (a) follows from the fact that $\Phi(t-1, t) = \mathbf{I}$. Note that when $t = 1$, it holds that

$$\sum_{r=1}^{t-1} \left(\alpha(r-1) \sum_{j=1}^{n-\phi} |\Phi_{ij}(t-1, r) - \pi_j(r)| |d_j(r-1)| \right) = 0.$$

From (22) and (23), the LHS of (21) can be upper bounded by

$$|y(t) - x_i(t)| \leq (n-\phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{t}{\nu} \rceil} + (n-\phi) L \sum_{r=1}^{t-1} \alpha(r-1) \gamma^{\lceil \frac{t-r}{\nu} \rceil} + 2\alpha(t-1)L.$$

The proof is complete. □

Proof of Lemma 7

Proof. Recall (14),

$$|y(t) - x_i(t)| \leq (n-\phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{t}{\nu} \rceil} + (n-\phi) L \sum_{r=1}^{t-1} \alpha(r-1) \gamma^{\lceil \frac{t-r}{\nu} \rceil} + 2\alpha(t-1)L.$$

Since $0 < \gamma \leq 1$ and $\lim_{t \rightarrow \infty} \alpha(t) = 0$, it is easy to see that the first term and the third term on the RHS of (14) both converge. In particular,

$$\lim_{t \rightarrow \infty} (n - \phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{t}{\nu} \rceil} = 0,$$

and

$$\lim_{t \rightarrow \infty} 2\alpha(t-1)L = 0.$$

Define

$$\ell(t) = \sum_{r=1}^t \alpha(r-1) \gamma^{\lceil \frac{t+1-r}{\nu} \rceil}.$$

For any $t \geq 1$, we have

$$\begin{aligned} \ell(t) &= \sum_{r=1}^t \alpha(r-1) \gamma^{\lceil \frac{t+1-r}{\nu} \rceil} \\ &= \sum_{r=1}^{\lceil \frac{t}{2} \rceil} \alpha(r-1) \gamma^{\lceil \frac{t+1-r}{\nu} \rceil} + \sum_{r=\lceil \frac{t}{2} \rceil+1}^t \alpha(r-1) \gamma^{\lceil \frac{t+1-r}{\nu} \rceil} \\ &\leq \sum_{r=1}^{\lceil \frac{t}{2} \rceil} \alpha(r-1) \gamma^{\frac{t+1-r}{\nu}} + \sum_{r=\lceil \frac{t}{2} \rceil+1}^t \alpha(r-1) \gamma^{\frac{t+1-r}{\nu}} \\ &\leq \sum_{r=1}^{\lceil \frac{t}{2} \rceil} \alpha(0) \gamma^{\frac{t+1-r}{\nu}} + \alpha(\lceil \frac{t}{2} \rceil) \sum_{r=\lceil \frac{t}{2} \rceil+1}^t \gamma^{\frac{t+1-r}{\nu}} \\ &\leq \alpha(0) \frac{\gamma^{\frac{t}{2\nu}}}{1 - \gamma^{\frac{1}{\nu}}} + \frac{\alpha(\lceil \frac{t}{2} \rceil)}{1 - \gamma^{\frac{1}{\nu}}}. \end{aligned}$$

Thus, we get

$$\limsup_{t \rightarrow \infty} \ell(t) \leq \alpha(0) \lim_{t \rightarrow \infty} \frac{\gamma^{\frac{t}{2\nu}}}{1 - \gamma^{\frac{1}{\nu}}} + \lim_{t \rightarrow \infty} \frac{\alpha(\lceil \frac{t}{2} \rceil)}{1 - \gamma^{\frac{1}{\nu}}} = 0 + 0 = 0.$$

Taking limit sup on both sides of (14), we have

$$\begin{aligned} \limsup_{t \rightarrow \infty} |y(t) - x_i(t)| &\leq \lim_{t \rightarrow \infty} (n - \phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{t}{\nu} \rceil} + \limsup_{t \rightarrow \infty} (n - \phi) L \ell(t-1) + \lim_{t \rightarrow \infty} 2\alpha(t-1)L \\ &\leq 0 + 0 + 0 = 0. \end{aligned}$$

On the other hand, since $|y(t) - x_i(t)| \geq 0$ for each t , it holds that

$$\liminf_{t \rightarrow \infty} |y(t) - x_i(t)| \geq 0.$$

Thus,

$$\limsup_{t \rightarrow \infty} |y(t) - x_i(t)| \leq 0 \leq \liminf_{t \rightarrow \infty} |y(t) - x_i(t)|.$$

By definition, we know $\liminf_{t \rightarrow \infty} |y(t) - x_i(t)| \leq \limsup_{t \rightarrow \infty} |y(t) - x_i(t)|$.

Therefore, the limit of $|y(t) - x_i(t)|$ exists, and $\lim_{t \rightarrow \infty} |y(t) - x_i(t)| = 0$.

□

Proof of Lemma 8

Proof. Since $\sum_{j=1}^{n-\phi} \pi_i(t+1) = 1$, by Lemma 6, we have for all $i \in \mathcal{V} - \mathcal{F}$,

$$\begin{aligned}
& \sum_{j=1}^{n-\phi} \pi_i(t+1) |y(t) - x_j(t)| \\
& \leq \sum_{j=1}^{n-\phi} \pi_i(t+1) \left((n-\phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{t}{\nu} \rceil} + (n-\phi) L \sum_{r=1}^{t-1} \alpha(r-1) \gamma^{\lceil \frac{t-r}{\nu} \rceil} + 2\alpha(t-1)L \right) \\
& \leq (n-\phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{t}{\nu} \rceil} + (n-\phi) L \sum_{r=1}^{t-1} \alpha(r-1) \gamma^{\lceil \frac{t-r}{\nu} \rceil} + 2\alpha(t-1)L.
\end{aligned}$$

Using the inequality that for each r and t

$$\alpha(t)\alpha(r-1) \leq \frac{1}{2} (\alpha^2(t) + \alpha^2(r-1)),$$

we obtain

$$\begin{aligned}
& \sum_{t=2}^{\infty} \alpha(t) \sum_{j=1}^{n-\phi} \pi_i(t+1) |y(t) - x_j(t)| \\
& \leq \sum_{t=2}^{\infty} \left(\alpha(t) (n-\phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{t}{\nu} \rceil} + (n-\phi) L \sum_{r=1}^{t-1} \alpha(t)\alpha(r-1) \gamma^{\lceil \frac{t-r}{\nu} \rceil} + 2\alpha(t)\alpha(t-1)L \right) \\
& \leq \sum_{t=2}^{\infty} \alpha(t) (n-\phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{t}{\nu} \rceil} + \frac{(n-\phi)L}{2} \sum_{t=2}^{\infty} \sum_{r=1}^{t-1} \alpha^2(t) \gamma^{\lceil \frac{t-r}{\nu} \rceil} \\
& \quad + \frac{(n-\phi)L}{2} \sum_{t=2}^{\infty} \sum_{r=1}^{t-1} \alpha^2(r-1) \gamma^{\lceil \frac{t-r}{\nu} \rceil} + \sum_{t=2}^{\infty} (\alpha^2(t) + \alpha^2(t-1)) L.
\end{aligned} \tag{24}$$

To show $\sum_{t=2}^{\infty} \alpha(t) \sum_{j=1}^{n-\phi} \pi_i(t+1) |y(t) - x_j(t)| < \infty$, we show each of the four terms in the RHS of (24) is finite.

For the first term on the RHS of (24), we have

$$\begin{aligned}
& \sum_{t=2}^{\infty} \alpha(t) (n-\phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{t}{\nu} \rceil} = (n-\phi) \max\{|u|, |U|\} \sum_{t=2}^{\infty} \alpha(t) \gamma^{\lceil \frac{t}{\nu} \rceil} \\
& \stackrel{(a)}{\leq} (n-\phi) \max\{|u|, |U|\} \alpha(1) \sum_{t=2}^{\infty} \gamma^{\lceil \frac{t}{\nu} \rceil} \\
& \leq (n-\phi) \max\{|u|, |U|\} \alpha(1) \sum_{t=2}^{\infty} \gamma^{\frac{t}{\nu}} \\
& \leq (n-\phi) \max\{|u|, |U|\} \frac{\alpha(1)}{1 - \gamma^{\frac{1}{\nu}}} \\
& < \infty.
\end{aligned} \tag{25}$$

Inequality (a) holds due to the fact that $\alpha(t) \leq \alpha(1)$ for all $t \geq 1$.

For the second term in the RHS of (24), we have

$$\begin{aligned}
\frac{(n-\phi)L}{2} \sum_{t=2}^{\infty} \sum_{r=1}^{t-1} \alpha^2(t) \gamma^{\lceil \frac{t-r}{\nu} \rceil} &\leq \frac{(n-\phi)L}{2} \sum_{t=2}^{\infty} \sum_{r=1}^{t-1} \alpha^2(t) \gamma^{\frac{t-r}{\nu}} \\
&\leq \frac{(n-\phi)L}{2} \sum_{t=2}^{\infty} \alpha^2(t) \sum_{r=1}^{t-1} \gamma^{\frac{t-r}{\nu}} \\
&\leq \frac{(n-\phi)L}{2} \sum_{t=2}^{\infty} \alpha^2(t) \sum_{r=1}^{\infty} \gamma^{\frac{r}{\nu}} \\
&\leq \frac{(n-\phi)L}{2(1-\gamma^{\frac{1}{\nu}})} \sum_{t=2}^{\infty} \alpha^2(t) \quad \text{since} \quad \sum_{r=1}^{\infty} \gamma^{\frac{r}{\nu}} = \frac{\gamma^{\frac{1}{\nu}}}{1-\gamma^{\frac{1}{\nu}}} \leq \frac{1}{1-\gamma^{\frac{1}{\nu}}} \\
&< \infty \quad \text{due to the fact that} \quad \sum_{t=0}^{\infty} \alpha^2(t) < \infty
\end{aligned} \tag{26}$$

For the forth term in the RHS of (24), we get

$$\sum_{t=2}^{\infty} (\alpha^2(t) + \alpha^2(t-1)) L = L \sum_{t=2}^{\infty} \alpha^2(t) + L \sum_{t=2}^{\infty} \alpha^2(t-1) < \infty. \tag{27}$$

For the third term in the RHS of (24), for any fixed N , we get

$$\begin{aligned}
\frac{(n-\phi)L}{2} \sum_{t=2}^N \sum_{r=1}^{t-1} \alpha^2(r-1) \gamma^{\lceil \frac{t-r}{\nu} \rceil} &\leq \frac{(n-\phi)L}{2} \sum_{t=2}^N \sum_{r=1}^{t-1} \alpha^2(r-1) \gamma^{\frac{t-r}{\nu}} \\
&= \frac{(n-\phi)L}{2} \sum_{r=1}^{N-1} \alpha^2(r-1) \sum_{t=1}^{N-r} \gamma^{\frac{t}{\nu}} \\
&\leq \frac{(n-\phi)L}{2(1-\gamma^{\frac{1}{\nu}})} \sum_{r=1}^{N-1} \alpha^2(r-1).
\end{aligned}$$

Thus, we get

$$\frac{(n-\phi)L}{2} \sum_{t=2}^{\infty} \sum_{r=1}^{t-1} \alpha^2(r-1) \gamma^{\lceil \frac{t-r}{\nu} \rceil} \leq \frac{(n-\phi)L}{2(1-\gamma^{\frac{1}{\nu}})} \sum_{r=1}^{\infty} \alpha^2(r-1) < \infty. \tag{28}$$

In addition, for $t = 0$, it holds that $|y(0) - x_j(0)| \leq U - u$. For $t = 1$, by Lemma 6, we have

$$\sum_{j=1}^{n-\phi} \pi_i(2) |y(1) - x_j(1)| \leq (n-\phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{1}{\nu} \rceil} + 2\alpha(0)L.$$

Thus,

$$\begin{aligned}
\alpha(0) \sum_{j=1}^{n-\phi} \pi_i(1) |y(0) - x_j(0)| + \alpha(1) \sum_{j=1}^{n-\phi} \pi_i(2) |y(1) - x_j(1)| &\leq \alpha(0) (U - u) + 2\alpha(0)L \\
&\quad + \alpha(1)(n-\phi) \max\{|u|, |U|\} \gamma^{\lceil \frac{1}{\nu} \rceil} \\
&< \infty.
\end{aligned} \tag{29}$$

By (25), (26), (27), (28) and (29), we conclude that

$$\sum_{t=0}^{\infty} \alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| < \infty,$$

proving the lemma. □

Proof of Theorem 4

Proof. Recall that each $g_i(\cdot)$ is defined as

$$g_i(x) = \mathbf{A}_{1i}h_1(x) + \mathbf{A}_{2i}h_2(x) + \dots + \mathbf{A}_{ki}h_k(x),$$

for $i \in \mathcal{V}$, where $\mathbf{A}_{ji} \geq 0$ and $\sum_{j=1}^k \mathbf{A}_{ji} = 1$. Let $Y^i = \operatorname{argmin} g_i(x)$ and $Y_j^i = \operatorname{argmin} \mathbf{A}_{ji}h_j(x)$ for $j = 1, \dots, k$. Since for each $j \in \{1, \dots, k\}$ such that $\mathbf{A}_{ji} = 0$, $\operatorname{argmin} \mathbf{A}_{ji}h_j(x) = 0$ is a constant function over the whole real line, it holds that $Y_j^i = \mathbb{R}$. Since positive constant scaling does not affect the optimal set of a function, for each $j \in \{1, \dots, k\}$ such that $\mathbf{A}_{ji} > 0$, it holds that $Y_j^i = X_j$. In addition, because $h_1(x), \dots, h_k(x)$ are solution redundant functions, i.e., $\cap_{j=1}^k X_j \neq \emptyset$, functions $\mathbf{A}_{1i}h_1(x), \dots, \mathbf{A}_{ki}h_k(x)$ are also solution redundant. By Proposition 2 we have

$$Y^i = \cap_{j: \mathbf{A}_{ji} > 0} X_j \supseteq \cap_{j=1}^k X_j = X, \text{ for all } i \in \mathcal{V}.$$

Let $x' \in X$. Define g_j^* as the optimal value of function $g_j(\cdot)$ for each $j \in \mathcal{V}$. We have

$$\begin{aligned} |y(t+1) - x'|^2 &\leq |y(t) - x'|^2 + 4L\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| \\ &\quad - 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j(x')) + \alpha^2(t)(n-\phi)L^2 \\ &\stackrel{(a)}{=} |y(t) - x'|^2 + 4L\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| \\ &\quad - 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j^*) + \alpha^2(t)(n-\phi)L^2. \end{aligned} \tag{30}$$

Equality (a) holds because of $x' \in X \subseteq Y^j$ for each $j \in \mathcal{V}$, then $g_j(x') = g_j^*$.

For each $t \geq 0$, define

$$\begin{aligned} a_t &= |y(t) - x'|^2, \\ b_t &= 2\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j^*), \\ c_t &= 4L\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| + \alpha^2(t)(n-\phi)L^2. \end{aligned}$$

It is easy to see that $a_t \geq 0$ and $c_t \geq 0$ for each t . Since g_j^* is the optimal value of function $g_j(\cdot)$, it holds that $b_t \geq 0$ for each t . Thus, $\{a_t\}_{t=0}^\infty, \{b_t\}_{t=0}^\infty$ and $\{c_t\}_{t=0}^\infty$ are three non-negative sequences. By (30), it holds that

$$a_{t+1} \leq a_t - b_t + c_t \quad \text{for each } t \geq 0.$$

By Lemma 8, it holds that

$$4L \sum_{t=0}^{\infty} \alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| < \infty.$$

In addition, since $\sum_{t=0}^{\infty} \alpha^2(t) < \infty$, it holds that

$$(n - \phi)L^2 \sum_{t=0}^{\infty} \alpha^2(t) < \infty.$$

Thus, we get

$$\begin{aligned} \sum_{t=0}^{\infty} c_t &= \sum_{t=0}^{\infty} \left(4L\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| + \alpha^2(t)(n - \phi)L^2 \right) \\ &= 4L \sum_{t=0}^{\infty} \left(\alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) |y(t) - x_j(t)| \right) + (n - \phi)L^2 \sum_{t=0}^{\infty} \alpha^2(t) \\ &< \infty. \end{aligned} \tag{31}$$

Therefore, applying Lemma 4 to the sequences $\{a_t\}_{t=0}^\infty, \{b_t\}_{t=0}^\infty$ and $\{c_t\}_{t=0}^\infty$, we have that for any $x' \in X$, $a_t = |y(t) - x'|$ converges, and

$$\sum_{t=0}^{\infty} b_t = \sum_{t=0}^{\infty} \alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j^*) < \infty. \tag{32}$$

Since $|y(t) - x'|$ converges for any fixed $x' \in X$, by definition of sequence convergence and the dynamic of $y(t)$ in (12), it is easy to see that $y(t)$ also converges. Let $\lim_{t \rightarrow \infty} y(t) = y$. Next we show that $y \in X$.

By continuity of $h(\cdot)$, we have

$$\lim_{t \rightarrow \infty} h(y(t)) = h\left(\lim_{t \rightarrow \infty} y(t)\right) = h(y).$$

Equivalently, for any $\epsilon > 0$, there exists T such that for any $t \geq T$, it holds that

$$|h(y(t)) - h(y)| < \epsilon.$$

Suppose $y \notin X$, then $h(y) - h^* > 0$. Let $\epsilon_0 = \frac{h(y) - h^*}{2}$. Then there exists T_0 such that for any $t \geq T_0$, it holds that

$$|h(y(t)) - h(y)| < \epsilon_0. \tag{33}$$

Let $\mathcal{I}_{t+1} \subseteq \mathcal{V} - \mathcal{F}$ be the set of indices such that for each $j \in \mathcal{I}_{t+1}$, $\pi_j(t+1) \geq \beta^\nu$. As $G(\mathcal{V}, \mathcal{E})$ satisfies Condition 2, $|\mathcal{I}_{t+1}| \geq \max\{k', f+1\}$. Since $g_j(y(t)) - g_j^* \geq 0$ for all j , then

$$\begin{aligned}
\sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j^*) &\geq \sum_{j \in \mathcal{I}_{t+1}} \pi_j(t+1) (g_j(y(t)) - g_j^*) \\
&\geq \beta^\nu \sum_{j \in \mathcal{I}_{t+1}} (g_j(y(t)) - g_j^*) \\
&= \beta^\nu \sum_{j \in \mathcal{I}_{t+1}} \sum_{i=1}^k \mathbf{A}_{ij} (h_i(y(t)) - h_i^*) \\
&= \beta^\nu \sum_{i=1}^k \left(\sum_{j \in \mathcal{I}_{t+1}} \mathbf{A}_{ij} \right) (h_i(y(t)) - h_i^*) \\
&\geq k\beta^\nu C_2 (h(y(t)) - h^*), \tag{34}
\end{aligned}$$

where

$$C_2 = \min_{\mathcal{I} \subseteq \mathcal{V}: |\mathcal{I}| \geq \max\{k', f+1\}} \sum_{i \in \mathcal{I}} \mathbf{A}_{ij},$$

and the last inequality follows from the fact that $h_i(y(t)) - h_i^* \geq 0$. In addition, as $sp(\mathbf{A}) = k'$, then $\sum_{i \in \mathcal{I}} \mathbf{A}_{ij} > 0$ for every $\mathcal{I} \subseteq \mathcal{V}: |\mathcal{I}| \geq \max\{k', f+1\}$. Since \mathbf{A} is finite, C_2 is well-defined and $C_2 > 0$. The relation (34) can be further bounded as follows.

$$\begin{aligned}
\sum_{t=0}^{\infty} \alpha(t) \sum_{j=1}^{n-\phi} \pi_j(t+1) (g_j(y(t)) - g_j^*) &\geq \sum_{t=0}^{\infty} \alpha(t) k\beta^\nu C_2 (h(y(t)) - h^*) \\
&\geq \sum_{t=T_0}^{\infty} \alpha(t) k\beta^\nu C_2 (h(y(t)) - h^*) \quad \text{as } h(y(t)) - h^* \geq 0, \forall t \\
&\geq \sum_{t=T_0}^{\infty} \alpha(t) k\beta^\nu C_2 (h(y) - h^* - \epsilon_0) \quad \text{by (33)} \\
&= \sum_{t=T_0}^{\infty} \alpha(t) k\beta^\nu C_2 \epsilon_0 \quad \text{since } \epsilon_0 = \frac{h(y) - h^*}{2} \\
&= \infty \quad \text{since } \sum_{t=0}^{\infty} \alpha(t) = \infty.
\end{aligned}$$

This contradicts the fact that (32) holds. Thus, the assumption that $y \notin X$ does not hold, and $y \in X$.

Therefore, we conclude that $y \in X$. That is, there exists $x^* \in X$ such that $y = x^*$ and

$$\lim_{t \rightarrow \infty} |y(t) - x^*| = 0. \tag{35}$$

By triangle inequality, we have

$$|x_i(t) - x^*| \leq |x_i(t) - y(t)| + |y(t) - x^*|.$$

Then, by Lemma 7 and (35), we have

$$\limsup_{t \rightarrow \infty} |x_i(t) - x^*| \leq \lim_{t \rightarrow \infty} |x_i(t) - y(t)| + \lim_{t \rightarrow \infty} |y(t) - x^*| = 0.$$

On the other hand, $\liminf_{t \rightarrow \infty} |x_i(t) - x^*| \geq 0$. Thus, limit of $|x_i(t) - x^*|$ exists and

$$\lim_{t \rightarrow \infty} |x_i(t) - x^*| = 0,$$

proving Theorem 4.

□